CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76201 ANA GLORIA DE SOUZA BRITO
24CC1305 - 1 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

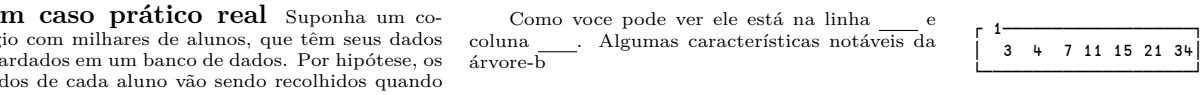
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

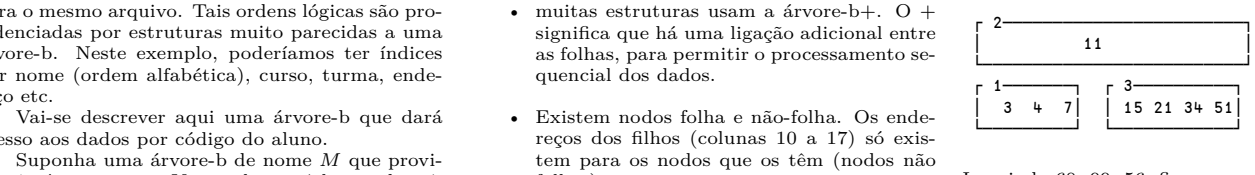
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

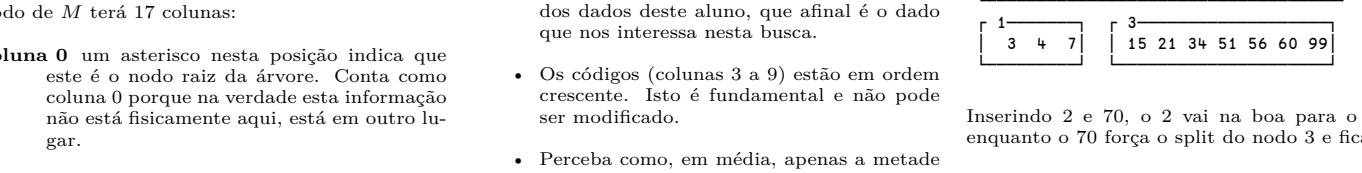


Agora, vamos inserir uma nova chave, o que vai provocar o split

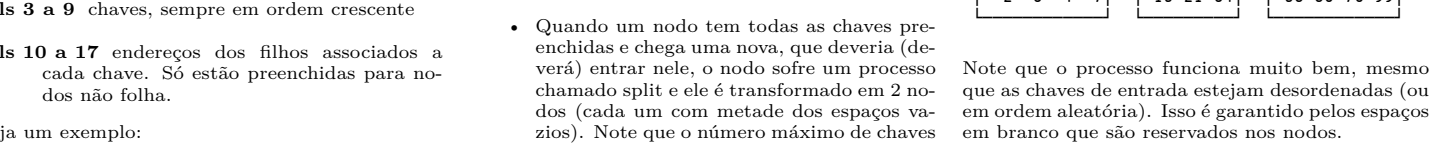
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 6   32   34   36   37   38   40    0
 2| 0 3   42   55   60    0    0    0    0  1 14  9 18
 3| 1 3  127  128  129    0    0    0    0
 4| 1 4   74   75   76   77    0    0    0
 5| 1 4  151  152  155  156    0    0    0
 6| 1 4  171  173  174  178    0    0    0
 7| 1 7  192  194  196  197  198  199  200
 8| 1 7   97   98  101  102  103  104  107
 9| 1 3   57   58   59    0    0    0    0
*10| 0 3   72  125  157    0    0    0    0  2 19 11 22
 11| 0 3  132  139  148    0    0    0    0  3 21 12  5
 12| 1 7  140  141  143  144  145  146  147
 13| 1 3   85   86   88    0    0    0    0
 14| 1 6   43   46   47   48   53   54    0
 15| 1 5  111  112  114  115  124    0    0
 16| 1 4  159  164  167  168    0    0    0
 17| 1 5  184  185  186  187  189    0    0
 18| 1 4   62   66   69   70    0    0    0
 19| 0 5   78   84   89   96  108    0    0  4 23 13 20  8 15
 20| 1 4   92   93   94   95    0    0    0
 21| 1 4  133  134  135  138    0    0    0
 22| 0 3  169  182  190    0    0    0    0 16  6 17  7
 23| 1 3   79   80   83    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 126 for incluido, qual linha,coluna na matriz?	Se o 117 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5   32   34   35   36   37    0    0
 2| 0 6   38   49   60   69   82   90    0  1 14  9  4 16  7 17
 3| 1 3  132  133  134    0    0    0    0
 4| 1 5   62   63   64   65   67    0    0
 5| 1 6  103  104  111  112  113  114    0
 6| 1 4  184  185  192  195    0    0    0
 7| 1 5   83   86   87   88   89    0    0
 8| 1 6  162  163  164  165  166  169    0
 9| 1 7   50   53   55   56   57   58   59
*10| 0 2  102  148    0    0    0    0    0  2 11 19
 11| 0 3  117  127  137    0    0    0    0  5 12  3 18
 12| 1 7  118  119  120  121  122  124  126
 13| 1 4  151  152  154  156    0    0    0
 14| 1 7   39   40   42   43   44   46   48
 15| 1 3  172  173  174    0    0    0    0
 16| 1 6   70   72   75   78   79   80    0
 17| 1 5   93   94   95   96   99    0    0
 18| 1 4  139  140  144  146    0    0    0
 19| 0 5  158  170  175  183  196    0    0 13  8 15 21  6 20
 20| 1 3  197  199  200    0    0    0    0
 21| 1 4  177  179  181  182    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 105 for incluido, qual linha,coluna na matriz?	Se o 189 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76218 ANDRE SANCHES
24CC1305 - 2 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

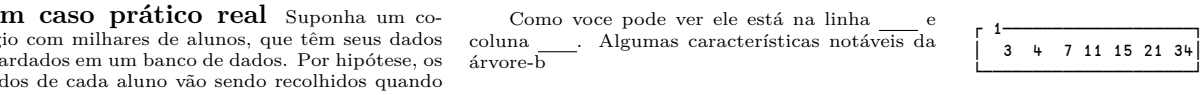
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

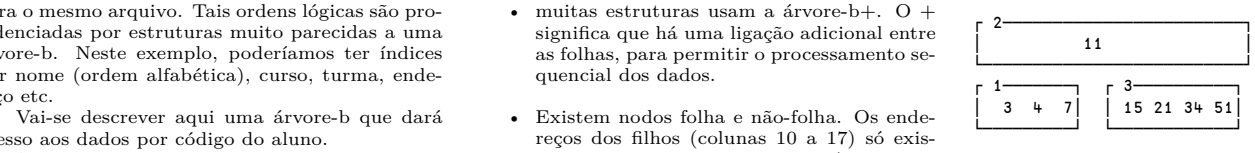
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

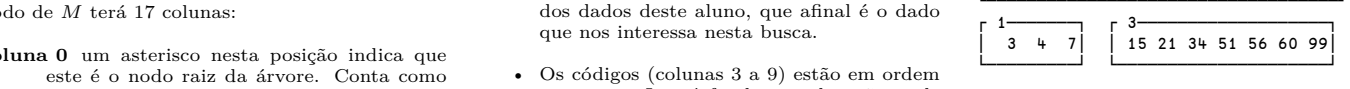


Agora, vamos inserir uma nova chave, o que vai provocar o split

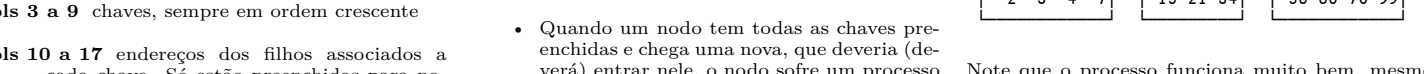
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 4    32   34   36   39    0    0    0
 2| 0 3    41   49   62    0    0    0    0  1 17  9 15
 3| 1 5   134  135  138  140  143    0    0
 4| 1 4    77   79   80   81    0    0    0
 5| 1 3   155  156  157    0    0    0    0
 6| 1 3    97  101  102    0    0    0    0
 7| 1 5   176  179  180  181  183    0    0
 8| 1 7   145  146  147  149  150  151  152
 9| 1 7    50   51   54   55   56   58   60
*10| 0 2    76  133    0    0    0    0    0  2 18 11
 11| 0 6   144  153  159  173  185  195    0  3  8  5 16  7 14 21
 12| 1 7    87   88   91   92   93   94   95
 13| 1 4   113  116  117  118    0    0    0
 14| 1 4   187  192  193  194    0    0    0
 15| 1 5    65   67   70   71   72    0    0
 16| 1 6   160  161  166  169  171  172    0
 17| 1 3    42   43   48    0    0    0    0
 18| 0 6    82   86   96  103  112  119    0  4 20 12  6 22 13 19
 19| 1 3   124  125  127    0    0    0    0
 20| 1 3    83   84   85    0    0    0    0
 21| 1 4   196  197  199  200    0    0    0
 22| 1 6   106  107  108  109  110  111    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 90 for incluido, qual linha,coluna na matriz?	Se o 162 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 3    35   36   37    0    0    0    0
 2| 0 6    38   44   54   63   81   94    0  1 17  9  4 22  8 15
 3| 1 3   119  123  126    0    0    0    0
 4| 1 4    56   57   59   60    0    0    0
 5| 1 5   146  149  151  152  153    0    0
 6| 1 3   103  104  105    0    0    0    0
 7| 1 3   171  174  179    0    0    0    0
 8| 1 5    84   85   86   87   92    0    0
 9| 1 7    46   48   49   50   51   52   53
*10| 0 2   102  145    0    0    0    0    0  2 11 18
 11| 0 4   106  118  127  138    0    0    0  6 19  3 16 13
 12| 1 7   192  194  195  196  197  199  200
 13| 1 5   140  141  142  143  144    0    0
 14| 1 4   156  157  158  159    0    0    0
 15| 1 6    95   96   97   98   99  101    0
 16| 1 6   128  131  132  134  135  136    0
 17| 1 5    39   40   41   42   43    0    0
 18| 0 5   154  160  168  184  191    0    0  5 14 21  7 20 12
 19| 1 5   107  109  112  114  116    0    0
 20| 1 4   185  187  189  190    0    0    0
 21| 1 3   162  163  167    0    0    0    0
 22| 1 5    65   68   72   74   80    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 88 for incluido, qual linha,coluna na matriz?	Se o 181 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76225 CASSIO COELHO PAZ
24CC1305 - 3 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

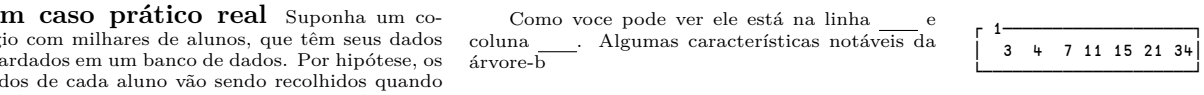
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

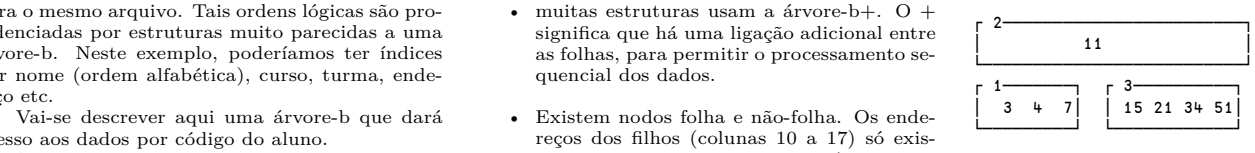
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

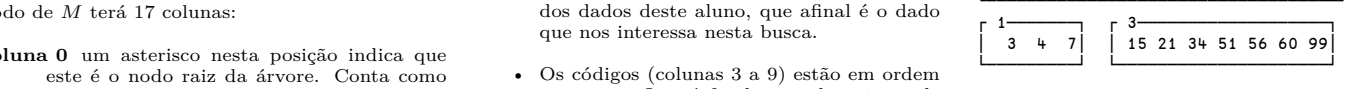


Agora, vamos inserir uma nova chave, o que vai provocar o split

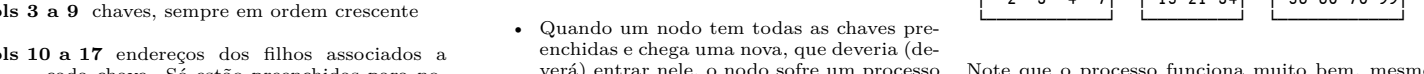
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 4    31   32   33   34    0    0    0
 2| 0 6    35   43   53   58   70   83    0  1  8 21 18  4 15  7
 3| 1 5   118  119  121  122  124    0    0
 4| 1 6    59   60   62   63   65   66    0
 5| 1 5   147  149  152  155  156    0    0
 6| 1 6    97   98   99  100  104  107    0
 7| 1 6    84   85   87   88   90   93    0
 8| 1 4    37   39   40   42    0    0    0
 9| 1 4   172  174  175  176    0    0    0
*10| 0 2    96  146    0    0    0    0    0  2 11 17
 11| 0 4   108  117  129  136    0    0    0  6 13  3 19 12
 12| 1 5   138  139  140  141  143    0    0
 13| 1 6   110  111  113  114  115  116    0
 14| 1 7   160  161  163  164  165  169  170
 15| 1 5    71   73   74   76   77    0    0
 16| 1 7   179  180  183  185  187  188  191
 17| 0 4   158  171  178  193    0    0    0  5 14  9 16 20
 18| 1 3    54   56   57    0    0    0    0
 19| 1 4   130  131  132  135    0    0    0
 20| 1 3   195  196  200    0    0    0    0
 21| 1 4    44   46   48   52    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 61 for incluido, qual linha,coluna na matriz?	Se o 49 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 4    32   35   38   39    0    0    0
 2| 0 5    40   51   60   73   86    0    0  1 24 12  6  9 20
 3| 1 6   123  124  125  126  127  130    0
 4| 1 5   180  181  182  183  184    0    0
 5| 1 4   151  157  158  159    0    0    0
 6| 1 7    61   65   66   67   68   69   71
 7| 1 4   103  104  105  106    0    0    0
 8| 1 4   186  187  188  189    0    0    0
 9| 1 5    74   75   77   80   84    0    0
*10| 0 3    92  122  171    0    0    0    0  2 18 11 22
 11| 0 4   131  138  149  160    0    0    0  3 15 21  5 23
 12| 1 4    53   54   56   59    0    0    0
 13| 1 4   115  116  119  121    0    0    0
 14| 1 5    93   94   95   96   99    0    0
 15| 1 4   133  134  135  137    0    0    0
 16| 1 4   173  174  175  176    0    0    0
 17| 1 4   109  110  111  112    0    0    0
 18| 0 3   100  108  113    0    0    0    0 14  7 17 13
 19| 1 3   194  195  196    0    0    0    0
 20| 1 4    87   88   89   90    0    0    0
 21| 1 3   143  147  148    0    0    0    0
 22| 0 3   179  185  193    0    0    0    0 16  4  8 19
 23| 1 4   162  165  168  170    0    0    0
 24| 1 4    42   45   47   48    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 62 for incluido, qual linha,coluna na matriz?	Se o 144 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76232 CAUA DOS SANTOS COSTA
24CC1305 - 4 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

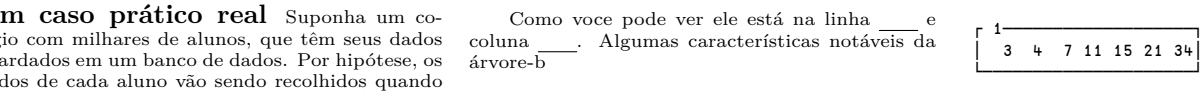
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

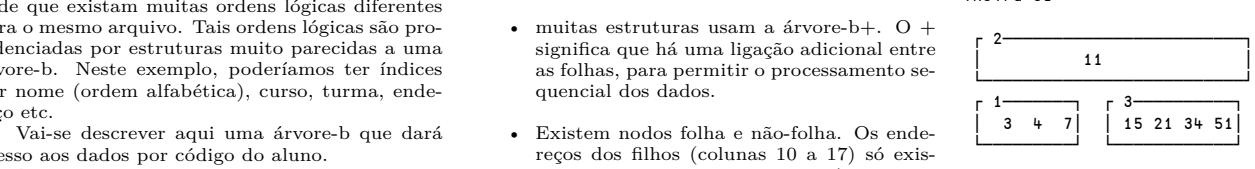
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

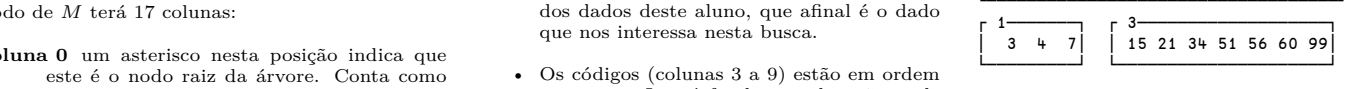


Agora, vamos inserir uma nova chave, o que vai provocar o split

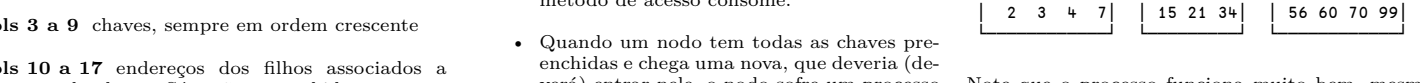
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 7   32   33   34   35   36   38   40
 2| 0 3   43   52   63    0    0    0    0  1 13  5  9
 3| 1 5   89   92   95   96   97    0    0
 4| 1 5  180  181  183  184  185    0    0
 5| 1 7   54   55   57   58   60   61   62
 6| 1 4  154  155  157  158    0    0    0
 7| 1 6  134  136  138  139  142  144    0
 8| 1 4  194  195  197  198    0    0    0
 9| 1 5   64   65   66   70   72    0    0
*10| 0 3   74  131  169    0    0    0    0  2 18 11 21
 11| 0 3  146  153  159    0    0    0    0  7 17  6 19
 12| 1 6  103  106  110  111  113  115    0
 13| 1 6   45   47   48   49   50   51    0
 14| 1 4  173  176  177  178    0    0    0
 15| 1 6  119  122  124  125  126  129    0
 16| 1 6   75   76   77   78   82   85    0
 17| 1 4  148  149  151  152    0    0    0
 18| 0 3   88   98  117    0    0    0    0 16  3 12 15
 19| 1 7  160  161  162  163  164  165  166
 20| 1 3  187  191  192    0    0    0    0
 21| 0 3  179  186  193    0    0    0    0 14  4 20  8
```

Responda agora:

Qual a altura da árvore B ?	Se o 56 for incluido, qual linha,coluna na matriz?	Se o 196 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5   31   32   34   35   36    0    0
 2| 0 5   37   44   50   57   64    0    0  1 16  6 22 12 20
 3| 1 5   76   78   80   86   87    0    0
 4| 1 4  131  132  135  137    0    0    0
 5| 1 7  172  174  175  176  177  178  179
 6| 1 4   45   46   47   49    0    0    0
 7| 1 3   98   99  104    0    0    0    0
 8| 1 3  140  141  143    0    0    0    0
 9| 1 7  192  193  194  195  197  198  199
*10| 0 2   71  129    0    0    0    0    0  2 18 11
 11| 0 6  139  146  154  167  181  190    0  4  8 13 21  5 14  9
 12| 1 4   58   59   60   61    0    0    0
 13| 1 5  148  149  150  152  153    0    0
 14| 1 5  185  186  187  188  189    0    0
 15| 1 5  117  122  124  125  126    0    0
 16| 1 4   38   39   40   41    0    0    0
 17| 1 5   89   91   93   94   96    0    0
 18| 0 4   88   97  106  115    0    0    0  3 17  7 19 15
 19| 1 5  108  109  110  112  114    0    0
 20| 1 3   66   67   68    0    0    0    0
 21| 1 5  157  159  160  161  165    0    0
 22| 1 4   51   52   54   56    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 133 for incluido, qual linha,coluna na matriz?	Se o 162 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76249 CAUE DAS NEVES MACHADO
24CC1305 - 5 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

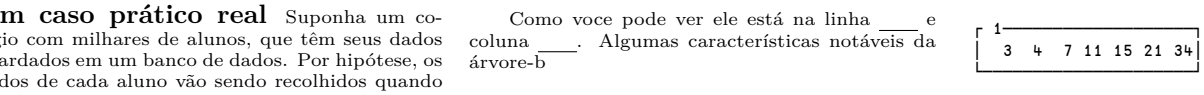
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

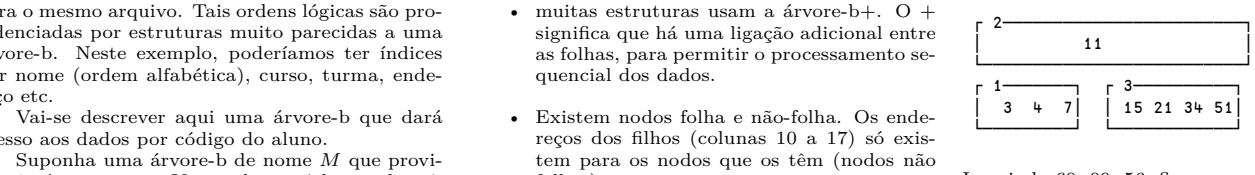
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

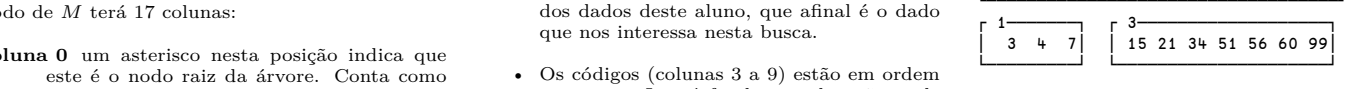


Agora, vamos inserir uma nova chave, o que vai provocar o split

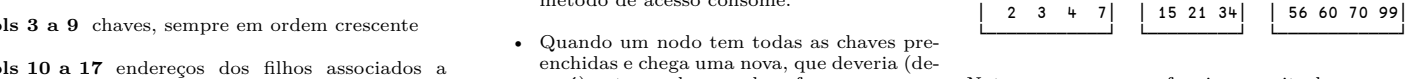
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   32   33   34   35   36    0    0
 2| 0  3   37   48   59    0    0    0    0  1 16  4 13
 3| 1  4   97   98   99  101    0    0    0
 4| 1  6   50   51   53   54   55   58    0
 5| 1  6  150  153  154  155  156  159    0
 6| 1  4  110  112  115  118    0    0    0
 7| 1  4  171  175  176  177    0    0    0
 8| 1  5   88   89   91   92   95    0    0
 9| 1  3  184  185  186    0    0    0    0
*10| 0  3   68  108  147    0    0    0    0  2 17 11 22
11| 0  3  119  126  135    0    0    0    0  6 19 12 21
12| 1  5  127  128  130  132  133    0    0
13| 1  5   62   64   65   66   67    0    0
14| 1  4   74   77   78   79    0    0    0
15| 1  3  104  106  107    0    0    0    0
16| 1  3   39   43   46    0    0    0    0
17| 0  4   80   87   96  102    0    0    0  14 20  8  3 15
18| 1  5  162  164  166  168  169    0    0
19| 1  4  120  122  124  125    0    0    0
20| 1  4   82   83   84   85    0    0    0
21| 1  4  136  137  141  142    0    0    0
22| 0  5  160  170  178  183  187    0    0  5 18  7 24  9 23
23| 1  5  189  194  198  199  200    0    0
24| 1  3  179  180  182    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 149 for incluido, qual linha,coluna na matriz?	Se o 57 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   31   33   34   35   36    0    0
 2| 0  3   38   45   53    0    0    0    0  1 12 20  5
 3| 1  6  136  139  140  141  145  146    0
 4| 1  5   95   96   97  100  102    0    0
 5| 1  6   56   58   59   62   65   66    0
 6| 1  4  162  163  165  166    0    0    0
 7| 1  7  189  191  194  195  198  199  200
 8| 1  6   80   81   83   84   85   86    0
 9| 1  7  104  105  106  107  108  110  113
*10| 0  3   67  103  160    0    0    0    0  2 21 11 19
11| 0  4  116  124  135  147    0    0    0  9 15 22  3 13
12| 1  3   39   40   41    0    0    0    0
13| 1  7  148  149  151  152  153  156  159
14| 1  4  181  183  184  186    0    0    0
15| 1  3  118  119  122    0    0    0    0
16| 1  6   68   69   73   75   77   78    0
17| 1  3  174  176  178    0    0    0    0
18| 1  4   89   91   92   93    0    0    0
19| 0  3  172  179  187    0    0    0    0  6 17 14  7
20| 1  4   48   50   51   52    0    0    0
21| 0  3   79   87   94    0    0    0    0 16  8 18  4
22| 1  4  126  128  132  134    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 61 for incluido, qual linha,coluna na matriz?	Se o 175 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76256 CLEVERSON LUIZ VOITACH JUNIOR
24CC1305 - 6 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

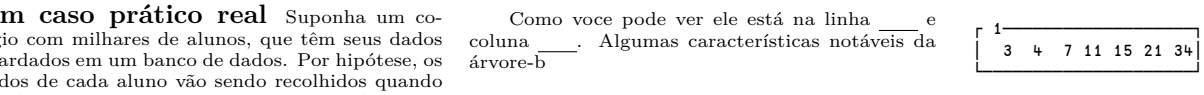
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

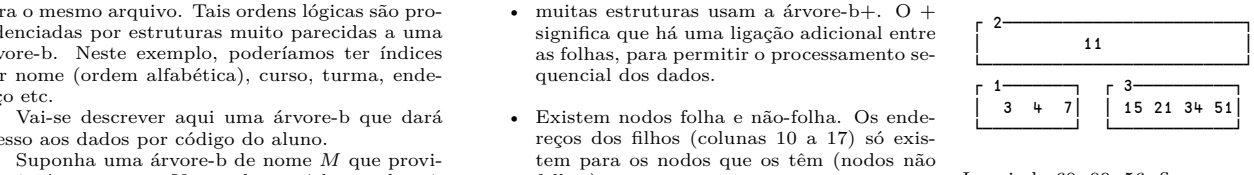
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

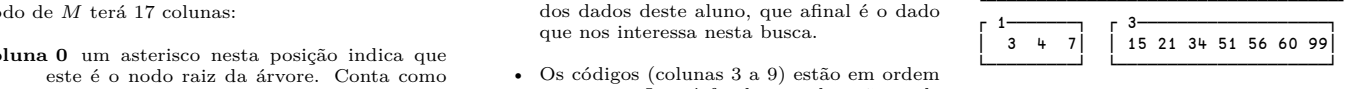


Agora, vamos inserir uma nova chave, o que vai provocar o split

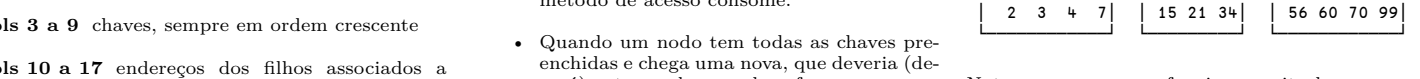
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 7   32  33  35  37  39  40  41
 2| 0 5   42  52  60  72  83   0   0  1  9 20  5 16  7
 3| 1 6  165 166 168 169 171 172   0
 4| 1 6   99 101 104 105 107 109   0
 5| 1 7   61  62  63  66  67  69  70
 6| 1 3  122 123 125   0   0   0   0
 7| 1 7   85  86  88  91  92  95  96
 8| 1 7  176 177 179 180 182 184 185
 9| 1 3   46  48  50   0   0   0   0
*10| 0 2   98 145   0   0   0   0   0  2 11 17
 11| 0 4  110 121 127 134   0   0   0  4 15  6 19 12
 12| 1 6  136 137 138 140 143 144   0
 13| 1 4  187 189 190 193   0   0   0
 14| 1 7  148 149 151 152 153 154 156
 15| 1 5  111 112 114 117 120   0   0
 16| 1 4   73  74  75  80   0   0   0
 17| 0 4  158 174 186 195   0   0   0 14  3  8 13 18
 18| 1 4  197 198 199 200   0   0   0
 19| 1 5  129 130 131 132 133   0   0
 20| 1 4   53  56  57  58   0   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 36 for incluido, qual linha,coluna na matriz?	Se o 167 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 4   33  34  35  38   0   0   0
 2| 0 6   40  50  58  63  74  81   0  1  8 20  6 15  4 14
 3| 1 3  114 116 122   0   0   0   0
 4| 1 5   75  77  78  79  80   0   0
 5| 1 4  168 169 170 171   0   0   0
 6| 1 3   59  60  61   0   0   0   0
 7| 1 4   98  99 100 102   0   0   0
 8| 1 4   41  42  46  48   0   0   0
 9| 1 4  136 139 140 142   0   0   0
*10| 0 2   96 158   0   0   0   0   0  2 11 18
 11| 0 6  106 113 123 132 143 148   0  7 19  3 21  9 23 16
 12| 1 7  186 187 188 189 190 193 194
 13| 1 4  160 161 162 163   0   0   0
 14| 1 6   85  88  90  92  93  94   0
 15| 1 5   64  65  67  70  73   0   0
 16| 1 3  149 150 154   0   0   0   0
 17| 1 6  174 175 179 180 181 183   0
 18| 0 4  165 172 184 195   0   0   0 13  5 17 12 22
 19| 1 4  107 109 110 112   0   0   0
 20| 1 5   51  52  55  56  57   0   0
 21| 1 5  124 127 129 130 131   0   0
 22| 1 3  196 198 200   0   0   0   0
 23| 1 3  144 145 146   0   0   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 76 for incluido, qual linha,coluna na matriz?	Se o 43 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76263 DAVI HONORIO ROSA
24CC1305 - 7 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

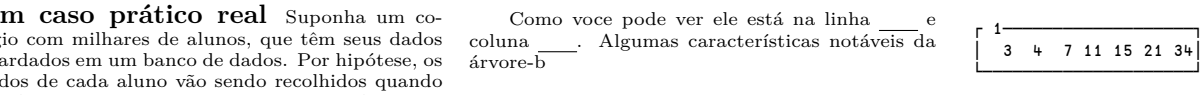
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

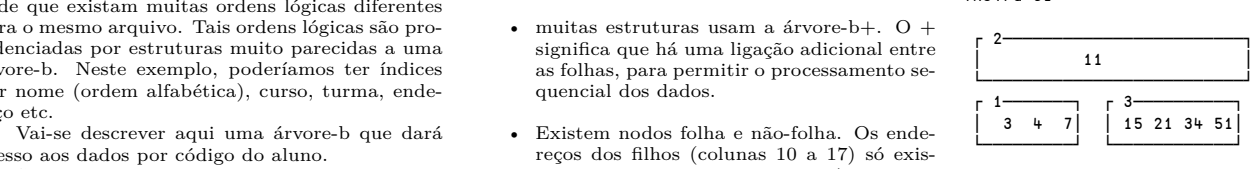
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

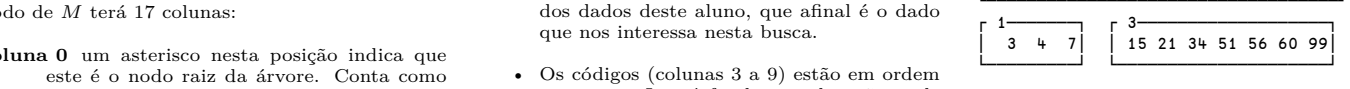


Agora, vamos inserir uma nova chave, o que vai provocar o split

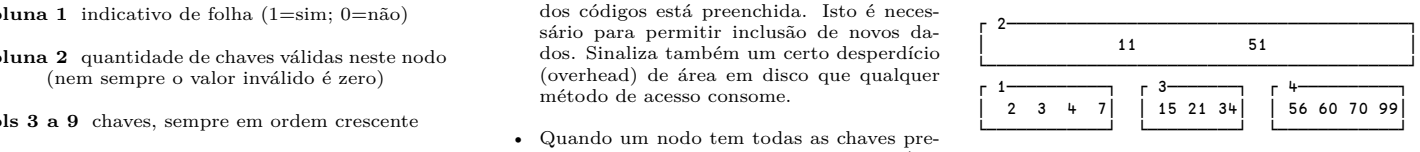
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   34   36   39   40   41    0    0
 2| 0  3   42   50   54    0    0    0    0  1 16 14  6
 3| 1  3   75   78   80    0    0    0    0
 4| 1  6  152  153  155  156  157  159    0
 5| 1  4  172  174  176  180    0    0    0
 6| 1  4   55   57   58   59    0    0    0
 7| 1  4  128  129  131  138    0    0    0
 8| 1  5  111  114  115  117  119    0    0
 9| 1  6  193  194  195  196  198  199    0
*10| 0  2   61  127    0    0    0    0    0  2 18 11
 11| 0  6  139  151  160  171  181  190    0  7 20  4 13  5 15  9
 12| 1  7   95   97   99  100  105  106  107
 13| 1  7  163  164  165  166  168  169  170
 14| 1  3   51   52   53    0    0    0    0
 15| 1  5  184  186  187  188  189    0    0
 16| 1  5   43   44   45   46   48    0    0
 17| 1  4   63   65   67   68    0    0    0
 18| 0  6   70   74   83   94  109  121    0 17 22  3 19 12  8 21
 19| 1  4   85   86   88   90    0    0    0
 20| 1  5  140  141  143  145  146    0    0
 21| 1  3  124  125  126    0    0    0    0
 22| 1  3   71   72   73    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 96 for incluido, qual linha,coluna na matriz?	Se o 116 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   31   32   36   37   38    0    0
 2| 0  6   40   62   71   82   90  103    0  1 12 15  7 13  5  4
 3| 1  3  172  173  174    0    0    0    0
 4| 1  7  105  106  107  109  112  114  119
 5| 1  7   91   93   94   96   97   98   99
 6| 1  4  122  123  124  127    0    0    0
 7| 1  7   72   73   74   76   77   78   80
 8| 1  5  182  184  188  190  191    0    0
 9| 1  5  152  153  154  155  160    0    0
*10| 0  2  120  170    0    0    0    0    0  2 11 19
 11| 0  4  129  141  150  161    0    0    0  6 20 16  9 17
 12| 1  7   44   45   47   49   51   54   60
 13| 1  5   83   84   85   87   88    0    0
 14| 1  5  193  194  195  196  200    0    0
 15| 1  5   63   64   65   66   70    0    0
 16| 1  7  142  143  145  146  147  148  149
 17| 1  5  162  164  166  168  169    0    0
 18| 1  4  176  177  178  179    0    0    0
 19| 0  3  175  181  192    0    0    0    0  3 18  8 14
 20| 1  4  136  137  138  139    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 110 for incluido, qual linha,coluna na matriz?	Se o 144 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76270 DORA DOS SANTOS PELLANDA
24CC1305 - 8 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

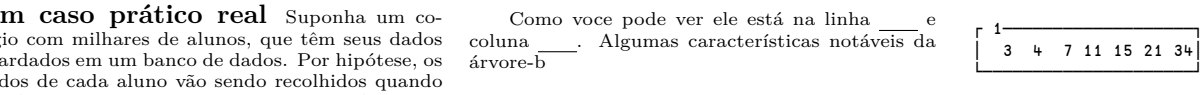
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

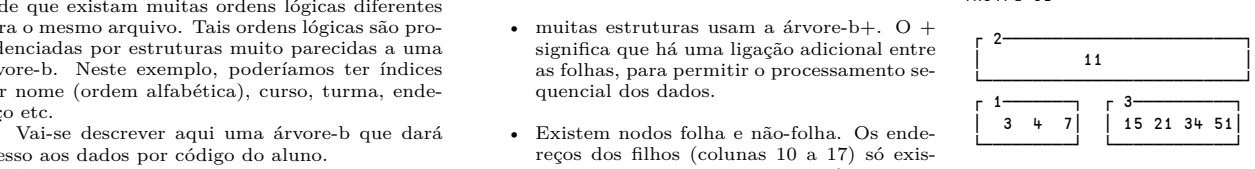
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

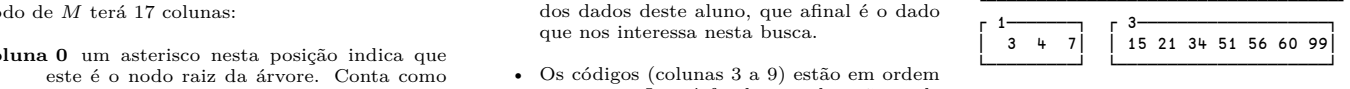


Agora, vamos inserir uma nova chave, o que vai provocar o split

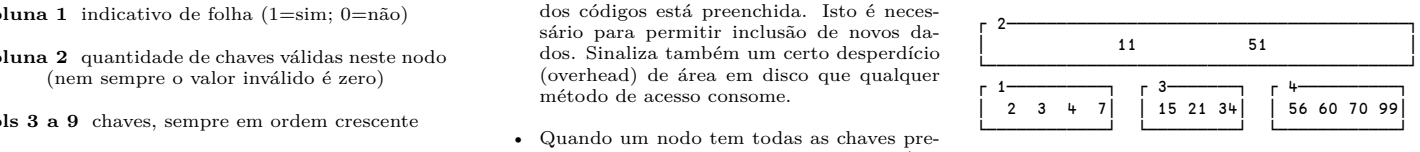
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-----------------------------------------------------------------------------------
 1|  1  4    31    33    34    36     0     0     0
 2|  0  3    37    44    57     0     0     0     0   1  18  15   7
 3|  1  7   138   140   142   143   144   145   146
 4|  1  5    80    82    86    87    89     0     0
 5|  1  6   168   169   170   171   172   173     0
 6|  1  7   118   120   121   124   126   127   128
 7|  1  6    60    64    65    66    67    68     0
 8|  1  3   189   190   191     0     0     0     0
 9|  1  4   105   106   110   111     0     0     0
*10|  0  2    69   117     0     0     0     0     0   2  19  11
11|  0  7   129   137   147   167   175   187   193   6  14   3  12   5  13   8  21
12|  1  6   148   150   153   160   162   165     0
13|  1  7   178   180   181   182   183   184   186
14|  1  4   131   132   135   136     0     0     0
15|  1  4    46    49    54    55     0     0     0
16|  1  5    70    71    73    77    78     0     0
17|  1  5    91    93    95    98    99     0     0
18|  1  4    38    39    42    43     0     0     0
19|  0  4    79    90   101   112     0     0     0  16   4  17   9  20
20|  1  3   114   115   116     0     0     0     0
21|  1  4   194   197   198   200     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 119 for incluido, qual linha,coluna na matriz?	Se o 176 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-----------------------------------------------------------------------------------
 1|  1  5    31    33    34    37    38     0     0
 2|  0  4    39    47    55    62     0     0     0   1  12   8  20  14
 3|  1  6   120   121   123   124   126   127     0
 4|  1  4   143   144   146   149     0     0     0
 5|  1  7   184   186   187   189   190   192   193
 6|  1  3    77    78    85     0     0     0     0
 7|  1  3   102   103   104     0     0     0     0
 8|  1  5    48    49    50    51    52     0     0
 9|  1  3   156   160   163     0     0     0     0
*10|  0  3    76   118   165     0     0     0     0   2  17  11  22
11|  0  3   133   142   150     0     0     0     0   3  15   4   9
12|  1  6    40    41    42    43    45    46     0
13|  1  3    87    89    90     0     0     0     0
14|  1  7    63    65    66    72    73    74    75
15|  1  4   134   138   140   141     0     0     0
16|  1  7   107   108   110   111   113   114   117
17|  0  4    86    92   100   106     0     0     0   6  13  23   7  16
18|  1  4   166   170   173   177     0     0     0
19|  1  3   197   198   199     0     0     0     0
20|  1  4    57    58    59    60     0     0     0
21|  1  3   180   181   182     0     0     0     0
22|  0  3   179   183   196     0     0     0     0  18  21   5  19
23|  1  6    93    94    95    97    98    99     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 53 for incluido, qual linha,coluna na matriz?	Se o 136 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76287 ENZO KAUAN CARVALHO
24CC1305 - 9 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

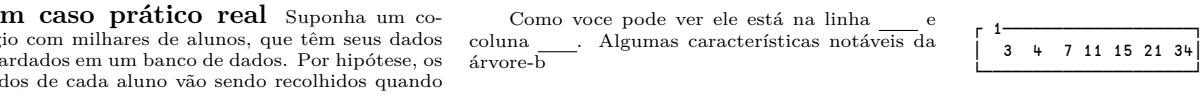
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

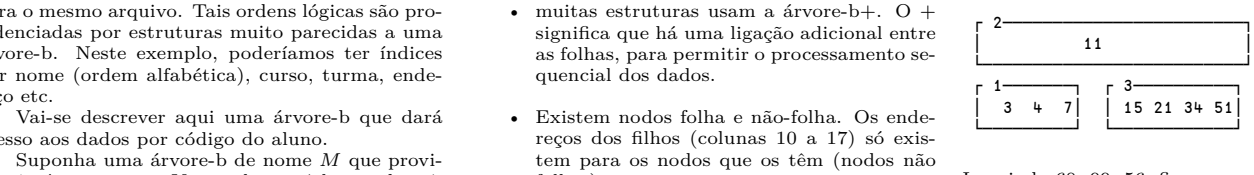
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

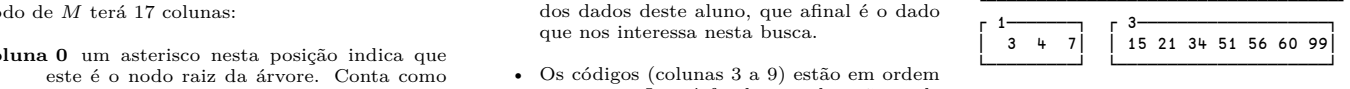


Agora, vamos inserir uma nova chave, o que vai provocar o split

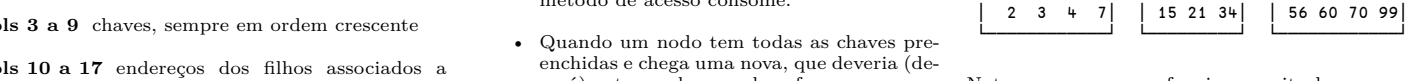
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5    32   34   35   36   38    0    0
 2| 0 3    39   50   66    0    0    0    0  1 14  9 15
 3| 1 7   165  166  167  169  170  174  175
 4| 1 5   104  107  108  109  110    0    0
 5| 1 6   114  117  118  122  130  131    0
 6| 1 4    75   77   78   80    0    0    0
 7| 1 5   183  184  185  187  188    0    0
 8| 1 6   156  158  159  160  161  162    0
 9| 1 7    52   53   54   55   60   61   65
*10| 0 3    74  111  164    0    0    0    0  2 21 11 20
 11| 0 3   132  147  155    0    0    0    0  5 12 17  8
 12| 1 6   137  138  142  144  145  146    0
 13| 1 5    90   91   92   94   95    0    0
 14| 1 5    40   41   43   46   49    0    0
 15| 1 5    67   68   70   71   73    0    0
 16| 1 3   179  180  181    0    0    0    0
 17| 1 4   149  150  152  154    0    0    0
 18| 1 3   196  197  199    0    0    0    0
 19| 1 5    82   83   84   85   87    0    0
 20| 0 4   178  182  189  193    0    0    0  3 16  7 22 18
 21| 0 3    81   88   98    0    0    0    0  6 19 13  4
 22| 1 3   190  191  192    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 63 for incluido, qual linha,coluna na matriz?	Se o 121 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 7    31   32   33   34   35   36   37
 2| 0 4    38   49   56   72    0    0    0  1 15 24  6 18
 3| 1 3   102  103  107    0    0    0    0
 4| 1 6   153  155  157  158  159  160    0
 5| 1 5   174  175  176  179  181    0    0
 6| 1 7    60   62   63   64   67   70   71
 7| 1 4   133  134  135  136    0    0    0
 8| 1 5    81   85   88   89   90    0    0
 9| 1 3   188  189  190    0    0    0    0
*10| 0 3    79  131  172    0    0    0    0  2 19 11 21
 11| 0 4   137  148  152  162    0    0    0  7 22 17  4 14
 12| 1 5   125  126  128  129  130    0    0
 13| 1 4   183  184  185  186    0    0    0
 14| 1 3   164  166  171    0    0    0    0
 15| 1 5    41   42   43   45   48    0    0
 16| 1 3    93   97   98    0    0    0    0
 17| 1 3   149  150  151    0    0    0    0
 18| 1 4    74   75   76   78    0    0    0
 19| 0 4    91  101  110  120    0    0    0  8 16  3 23 12
 20| 1 5   192  194  196  199  200    0    0
 21| 0 3   182  187  191    0    0    0    0  5 13  9 20
 22| 1 3   139  142  147    0    0    0    0
 23| 1 4   113  115  117  119    0    0    0
 24| 1 3    51   52   55    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 178 for incluido, qual linha,coluna na matriz?	Se o 177 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76294 FABRICIO JAWORSKI FERREIRA
24CC1305 - 10 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

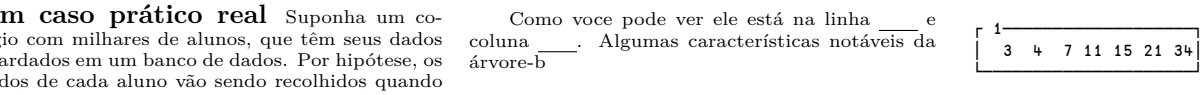
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

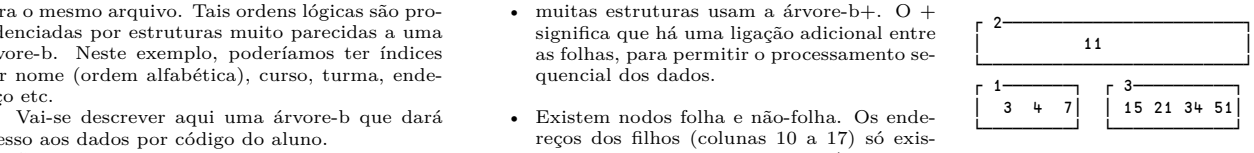
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

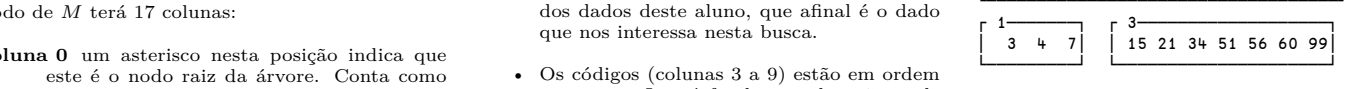


Agora, vamos inserir uma nova chave, o que vai provocar o split

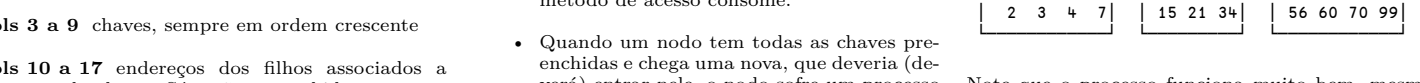
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 6    32   35   36   37   38   39    0
 2| 0 3    41   52   56    0    0    0    0  1 15  7 14
 3| 1 6   114  115  117  119  120  121    0
 4| 1 6   153  154  156  157  159  161    0
 5| 1 6    71   73   74   76   77   78    0
 6| 1 5   180  183  185  186  187    0    0
 7| 1 3    53   54   55    0    0    0    0
 8| 1 4    94   96   99  105    0    0    0
 9| 1 4   163  164  168  169    0    0    0
*10| 0 3    62  113  162    0    0    0    0  2 21 11 19
 11| 0 3   122  133  151    0    0    0    0  3 17 12  4
 12| 1 7   136  138  142  143  144  145  146
 13| 1 7    81   83   85   87   89   90   92
 14| 1 4    58   59   60   61    0    0    0
 15| 1 5    42   44   45   46   49    0    0
 16| 1 4   172  173  174  176    0    0    0
 17| 1 5   123  125  126  129  131    0    0
 18| 1 5   192  194  195  199  200    0    0
 19| 0 3   171  179  191    0    0    0    0  9 16  6 18
 20| 1 4    64   66   68   69    0    0    0
 21| 0 4    70   79   93  106    0    0    0 20  5 13  8 22
 22| 1 3   107  111  112    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 100 for incluido, qual linha,coluna na matriz?	Se o 72 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 7    32   34   35   36   37   39   41
 2| 0 4    43   50   56   70    0    0    0  1 22 13  5 14
 3| 1 5   123  124  125  126  129    0    0
 4| 1 6   158  161  162  166  169  170    0
 5| 1 7    57   58   59   60   61   68   69
 6| 1 6    93   94   95   97   99  100    0
 7| 1 5   181  182  183  185  186    0    0
 8| 1 4   132  133  134  135    0    0    0
 9| 1 7   108  109  112  113  117  118  120
*10| 0 3    81  122  157    0    0    0    0  2 18 11 21
 11| 0 3   131  136  149    0    0    0    0  3  8 15 20
 12| 1 4   174  176  177  178    0    0    0
 13| 1 4    51   52   53   55    0    0    0
 14| 1 5    71   72   75   76   80    0    0
 15| 1 5   139  140  142  143  144    0    0
 16| 1 4    82   85   86   87    0    0    0
 17| 1 3   103  104  105    0    0    0    0
 18| 0 3    92  101  106    0    0    0    0 16  6 17  9
 19| 1 3   194  195  200    0    0    0    0
 20| 1 6   150  151  152  154  155  156    0
 21| 0 3   172  180  189    0    0    0    0  4 12  7 19
 22| 1 3    44   45   49    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 167 for incluido, qual linha,coluna na matriz?	Se o 160 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76306 FELIPE FERNANDES DOS SANTOS
24CC1305 - 11 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

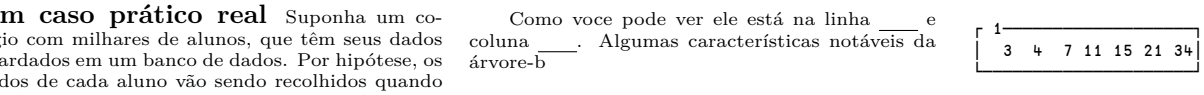
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

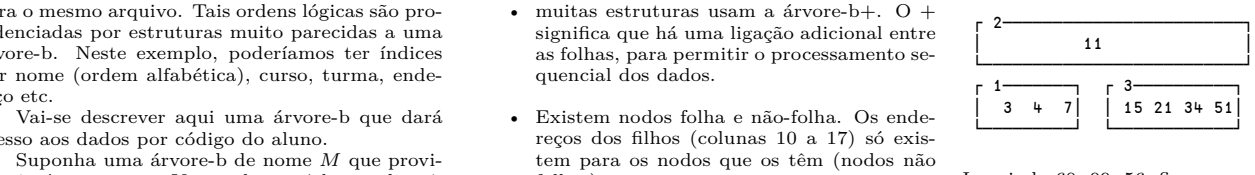
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

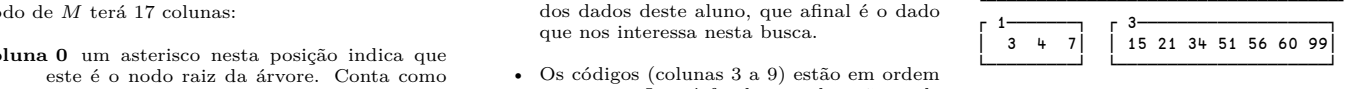


Agora, vamos inserir uma nova chave, o que vai provocar o split

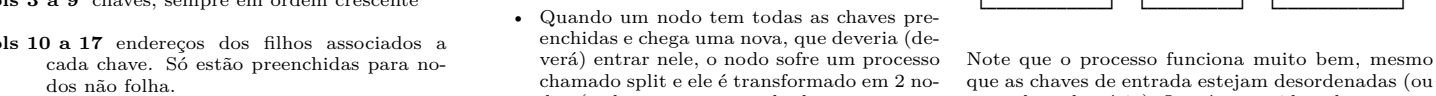
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 7   31  34  35  36  37  38  40
 2| 0 6   41  54  60  66  75  86   0  1  7 14 20  4 13  5
 3| 1 6  119 120 121 123 124 125   0
 4| 1 7   67  68  69  71  72  73  74
 5| 1 6   89  90  91  94  95  96   0
 6| 1 6  167 168 171 174 175 176   0
 7| 1 7   42  46  47  48  49  50  52
 8| 1 4  147 149 150 152   0   0   0
 9| 1 5  101 102 103 104 107   0   0
*10| 0 2   99 154   0   0   0   0   0  2 11 18
 11| 0 4  109 117 133 146   0   0   0  9 19  3 15  8
 12| 1 3  180 184 185   0   0   0   0
 13| 1 6   76  77  78  79  81  82   0
 14| 1 3   55  56  58   0   0   0   0
 15| 1 7  134 135 137 139 143 144 145
 16| 1 6  191 193 195 196 197 199   0
 17| 1 3  158 159 164   0   0   0   0
 18| 0 3  165 179 189   0   0   0   0 17  6 12 16
 19| 1 4  111 114 115 116   0   0   0
 20| 1 5   61  62  63  64  65   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 151 for incluido, qual linha,coluna na matriz?	Se o 92 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 6   31  33  35  36  37  38   0
 2| 0 6   42  53  70  84  90  97   0  1 20 13  6 18 12 19
 3| 1 6  146 147 148 149 151 152   0
 4| 1 3  109 110 113   0   0   0   0
 5| 1 7  181 182 184 185 186 188 189
 6| 1 4   73  74  78  82   0   0   0
 7| 1 7  133 137 138 140 142 143 144
 8| 1 3  193 194 195   0   0   0   0
 9| 1 5  155 158 162 163 164   0   0
*10| 0 2  108 145   0   0   0   0   0  2 17 11
 11| 0 5  154 170 179 191 196   0   0  3  9 15  5  8 21
 12| 1 5   91  92  94  95  96   0   0
 13| 1 7   54  59  60  61  62  64  65
 14| 1 5  115 118 120 121 122   0   0
 15| 1 4  171 174 177 178   0   0   0
 16| 1 4  127 128 129 130   0   0   0
 17| 0 3  114 125 131   0   0   0   0  4 14 16  7
 18| 1 3   86  87  88   0   0   0   0
 19| 1 5  100 102 103 104 107   0   0
 20| 1 6   45  47  48  49  50  52   0
 21| 1 4  197 198 199 200   0   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 75 for incluido, qual linha,coluna na matriz?	Se o 150 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76313 GABRIEL BORDIGNON
24CC1305 - 12 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	5	3	6	7	8	10	0	0								
2	0	4	11	20	32	44	0	0	0	1	20	15	7	12			
3	1	4	128	131	132	134	0	0	0								
4	1	5	149	150	151	153	160	0	0								
5	1	6	65	73	76	79	80	83	0								
6	1	5	182	183	188	189	191	0	0								
7	1	3	34	35	37	0	0	0	0								
8	1	4	106	109	111	114	0	0	0								
9	1	6	136	137	138	140	143	145	0								
*10	0	2	64	127	0	0	0	0	0	2	17	11					
11	0	6	135	146	162	171	181	192	0	3	9	4	14	21	6	18	
12	1	7	46	49	50	55	56	57	63								
13	1	4	88	89	91	94	0	0	0								
14	1	5	164	166	167	168	169	0	0								
15	1	6	21	22	23	28	29	30	0								
16	1	7	117	118	120	122	123	125	126								
17	0	4	84	95	105	116	0	0	0	5	13	19	8	16			
18	1	5	193	195	197	199	200	0	0								
19	1	5	96	98	100	102	103	0	0								
20	1	4	14	15	17	19	0	0	0								
21	1	3	174	175	177	0	0	0	0								

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	5	3	6	7	8	10	0	0								
2	0	4	11	20	32	44	0	0	0	1	20	15	7	12			
3	1	4	128	131	132	134	0	0	0								
4	1	5	149	150	151	153	160	0	0								
5	1	6	65	73	76	79	80	83	0								
6	1	6	182	183(187)	188	189	191	0									
7	1	3	34	35	37	0	0	0	0								
8	1	4	106	109	111	114	0	0	0								
9	1	6	136	137	138	140	143	145	0								
*10	0	2	64	127	0	0	0	0	0	2	17	11					
11	0	6	135	146	162	171	181	192	0	3	9	4	14	21	6	18	
12	1	7	46	49	50	55	56	57	63								
13	1	4	88	89	91	94	0	0	0								
14	1	5	164	166	167	168	169	0	0								
15	1	6	21	22	23	28	29	30	0								
16	1	7	117	118	120	122	123	125	126								
17	0	4	84	95	105	116	0	0	0	5	13	19	8	16			
18	1	5	193	195	197	199	200	0	0								
19	1	5	96	98	100	102	103	0	0								
20	1	4	14	15	17	19	0	0	0								
21	1	3	174	175	177	0	0	0	0								

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

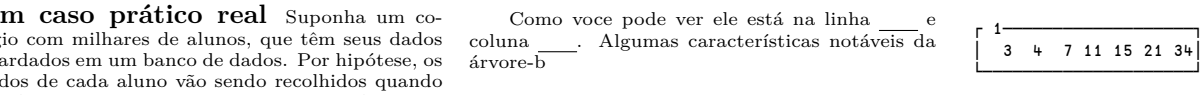
Algoritmo de Inclusão em árvore B

1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4: X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7: faça *split* do nodo X
8: INCLUA-BTREE (K)
9: abandone
10: senão
11: inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

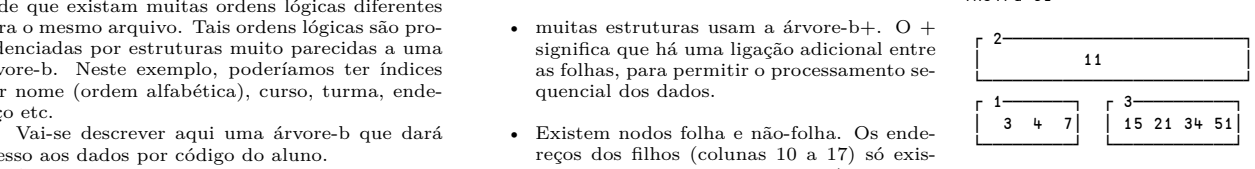
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

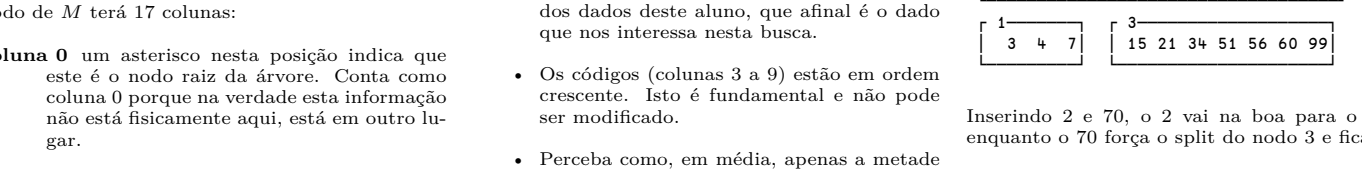


Agora, vamos inserir uma nova chave, o que vai provocar o split

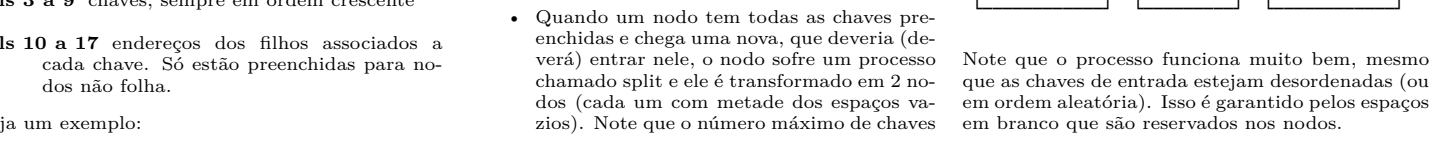
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  5    32    33    36    37    38     0     0
 2 | 0  3    39    55    62     0     0     0     0   1  17  12  15
 3 | 1  4   112   113   115   117     0     0     0
 4 | 1  4   150   151   153   154     0     0     0
 5 | 1  4    81    82    85    87     0     0     0
 6 | 1  3   175   176   177     0     0     0     0
 7 | 1  5   101   102   107   108   109     0     0
 8 | 1  5   138   139   140   141   142     0     0
 9 | 1  3   127   128   129     0     0     0     0
*10| 0  3    79   126   158     0     0     0     0   2  21  11  20
11 | 0  4   130   137   143   148     0     0     0   9  23   8  18   4
12 | 1  4    56    58    60    61     0     0     0
13 | 1  3   193   194   200     0     0     0     0
14 | 1  7   159   161   163   164   165   169   171
15 | 1  7    66    68    69    72    73    74    75
16 | 1  6   181   182   183   184   187   188     0
17 | 1  5    42    45    47    50    53     0     0
18 | 1  3   144   145   147     0     0     0     0
19 | 1  7    89    91    92    93    95    97    98
20 | 0  3   173   179   192     0     0     0     0  14   6  16  13
21 | 0  4    88   100   110   118     0     0     0   5  19   7   3  22
22 | 1  4   119   120   121   122     0     0     0
23 | 1  4   132   133   134   136     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 103 for incluido, qual linha,coluna na matriz?	Se o 111 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  7    31    33    34    35    36    39    41
 2 | 0  6    43    47    64    68    80    93     0   1   7  15  22   4  12   9
 3 | 1  7   105   106   107   108   110   111   112
 4 | 1  6    70    71    73    74    78    79     0
 5 | 1  3   142   143   144     0     0     0     0
 6 | 1  3   174   175   176     0     0     0     0
 7 | 1  3    44    45    46     0     0     0     0
 8 | 1  5   114   115   116   118   121     0     0
 9 | 1  5    95    98    99   100   101     0     0
*10| 0  2   102   154     0     0     0     0     0   2  11  18
11 | 0  5   113   122   130   141   146     0     0   3   8  16  20   5  21
12 | 1  6    81    82    84    86    87    90     0
13 | 1  6   190   191   192   193   196   197     0
14 | 1  5   155   156   157   162   164     0     0
15 | 1  4    49    52    56    59     0     0     0
16 | 1  3   123   124   127     0     0     0     0
17 | 1  3   166   167   168     0     0     0     0
18 | 0  4   165   170   177   189     0     0     0  14  17   6  19  13
19 | 1  6   178   179   181   182   183   184     0
20 | 1  4   132   134   137   139     0     0     0
21 | 1  4   147   148   149   151     0     0     0
22 | 1  3    65    66    67     0     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 186 for incluido, qual linha,coluna na matriz?	Se o 96 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76320 GEOVANNI TEIXEIRA ARAUJO
24CC1305 - 13 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

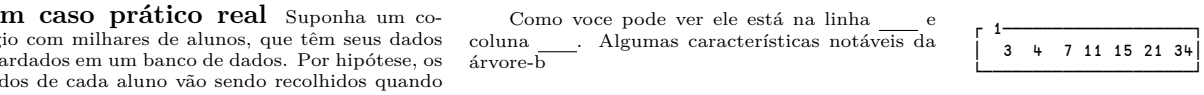
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

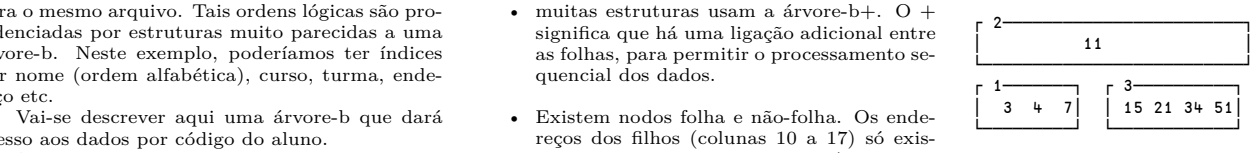
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

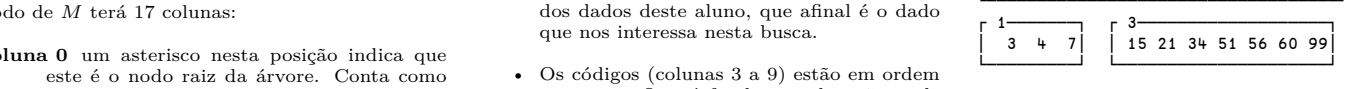


Agora, vamos inserir uma nova chave, o que vai provocar o split

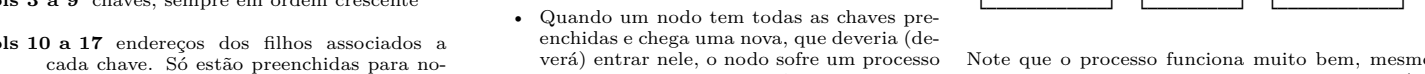
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 6    31   32   33   35   36   37    0
 2| 0 4    38   44   52   58    0    0    0  1 16  7 21 12
 3| 1 5    72   73   74   75   76    0    0
 4| 1 6   127  128  130  132  134  135    0
 5| 1 4    99  100  103  104    0    0    0
 6| 1 6   152  154  156  158  159  160    0
 7| 1 3    47   48   49    0    0    0    0
 8| 1 4   173  175  179  180    0    0    0
 9| 1 4   162  164  168  171    0    0    0
*10| 0 2    70  124    0    0    0    0    0  2 18 11
 11| 0 7   136  144  151  161  172  181  185  4 17 20  6  9  8 22 13
 12| 1 7    61   62   63   64   67   68   69
 13| 1 6   188  189  192  194  196  200    0
 14| 1 4   106  108  111  115    0    0    0
 15| 1 6    79   84   86   89   90   91    0
 16| 1 4    39   40   41   43    0    0    0
 17| 1 4   138  139  140  141    0    0    0
 18| 0 4    78   94  105  116    0    0    0  3 15  5 14 19
 19| 1 4   117  120  122  123    0    0    0
 20| 1 3   145  147  149    0    0    0    0
 21| 1 4    54   55   56   57    0    0    0
 22| 1 3   182  183  184    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 197 for incluido, qual linha,coluna na matriz?	Se o 155 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 6    31   34   35   38   39   40    0
 2| 0 5    41   46   51   63   72    0    0  1 12 20  6 13 19
 3| 1 5   140  147  150  151  152    0    0
 4| 1 5   119  120  121  122  124    0    0
 5| 1 4   162  164  166  169    0    0    0
 6| 1 6    54   55   56   57   60   62    0
 7| 1 6   154  155  156  157  158  159    0
 8| 1 6    80   81   86   87   91   94    0
 9| 1 4   182  183  185  186    0    0    0
*10| 0 2    79  139    0    0    0    0    0  2 16 11
 11| 0 5   153  160  171  180  188    0    0  3  7  5 21  9 17
 12| 1 4    42   43   44   45    0    0    0
 13| 1 5    64   67   69   70   71    0    0
 14| 1 6   104  107  111  112  114  117    0
 15| 1 7   128  129  130  131  134  136  137
 16| 0 4    97  103  118  127    0    0    0  8 18 14  4 15
 17| 1 5   189  190  197  198  200    0    0
 18| 1 4    98  100  101  102    0    0    0
 19| 1 4    73   74   76   78    0    0    0
 20| 1 3    47   48   50    0    0    0    0
 21| 1 4   173  175  176  178    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 177 for incluido, qual linha,coluna na matriz?	Se o 105 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76337 GUILHERME CAVALER KUSTER
24CC1305 - 14 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

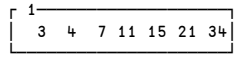
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

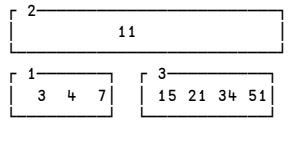
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

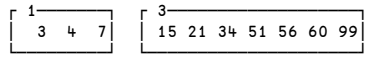


Agora, vamos inserir uma nova chave, o que vai provocar o split

```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica

Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
  +-------------------------------------------------------------------------------
  1| 1  7    35    37    38    39    40    43    44
  2| 0  3    46    66    74     0     0     0     0   1  15   9  14
  3| 1  4   153   155   156   157     0     0     0
  4| 1  5   113   114   115   116   117     0     0
  5| 1  3   192   193   194     0     0     0     0
  6| 1  4   133   134   136   139     0     0     0
  7| 1  3    93    94    99     0     0     0     0
  8| 1  5   184   186   188   189   190     0     0
  9| 1  5    68    69    70    71    73     0     0
*10| 0  2    82   132     0     0     0     0     0   2  17  11
 11| 0  7   140   151   159   166   183   191   195   6  20   3  19  13   8   5  21
 12| 1  6   120   122   124   125   126   127     0
 13| 1  7   167   168   170   174   176   178   179
 14| 1  4    77    78    80    81     0     0     0
 15| 1  7    49    50    51    54    55    57    58
 16| 1  5    83    85    86    90    91     0     0
 17| 0  4    92   100   109   118     0     0     0  16   7  18   4  12
 18| 1  6   101   102   103   106   107   108     0
 19| 1  3   160   162   165     0     0     0     0
 20| 1  6   141   142   144   147   149   150     0
 21| 1  4   196   197   199   200     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 177 for incluido, qual linha,coluna na matriz?	Se o 76 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
  +-------------------------------------------------------------------------------
  1| 1  5    32    33    34    35    36     0     0
  2| 0  4    37    47    65    70     0     0     0   1  13  16   5  23
  3| 1  4    90    91    92    93     0     0     0
  4| 1  3   121   123   126     0     0     0     0
  5| 1  4    66    67    68    69     0     0     0
  6| 1  5   148   149   151   152   153     0     0
  7| 1  4    75    77    78    79     0     0     0
  8| 1  5   102   105   107   108   109     0     0
  9| 1  3   171   173   179     0     0     0     0
*10| 0  3    74   100   147     0     0     0     0   2  19  11  21
 11| 0  4   110   117   128   136     0     0     0   8  15   4  22  12
 12| 1  6   138   141   142   143   144   146     0
 13| 1  5    39    42    43    44    45     0     0
 14| 1  5    95    96    97    98    99     0     0
 15| 1  4   113   114   115   116     0     0     0
 16| 1  7    54    55    56    58    61    63    64
 17| 1  5    81    83    84    85    88     0     0
 18| 1  4   156   157   160   162     0     0     0
 19| 0  3    80    89    94     0     0     0     0   7  17   3  14
 20| 1  7   181   183   187   192   198   199   200
 21| 0  3   154   168   180     0     0     0     0   6  18   9  20
 22| 1  4   130   132   134   135     0     0     0
 23| 1  3    71    72    73     0     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 178 for incluido, qual linha,coluna na matriz?	Se o 49 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76344 GUILHERME LUCIANO DE MELO
24CC1305 - 15 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

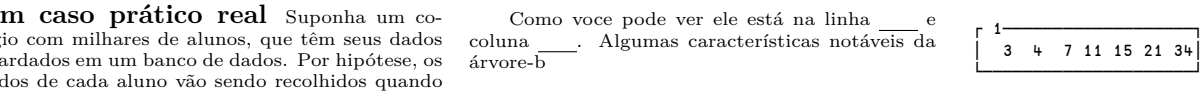
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

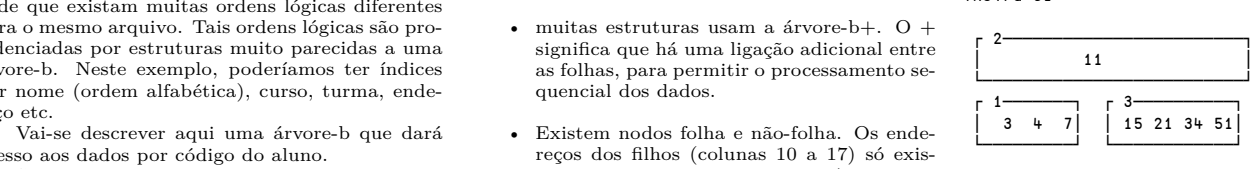
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

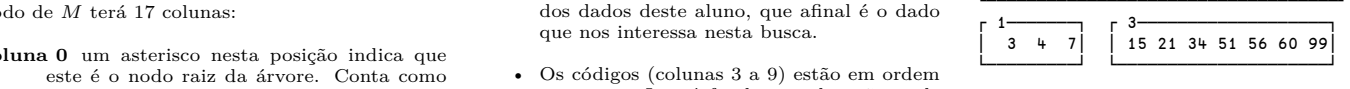


Agora, vamos inserir uma nova chave, o que vai provocar o split

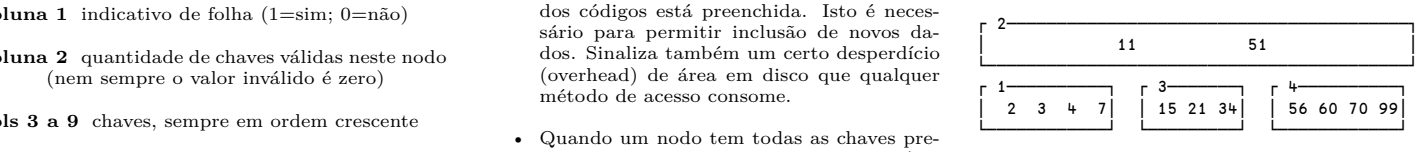
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
  +-------------------------------------------------------------------------------
  1| 1  7    32    37    38    40    45    46    47
  2| 0  5    48    56    66    76    81     0     0   1  16  14  18   4  22
  3| 1  3   142   143   144     0     0     0     0
  4| 1  3    77    78    79     0     0     0     0
  5| 1  5   117   119   120   123   124     0     0
  6| 1  4   168   170   172   174     0     0     0
  7| 1  4   197   198   199   200     0     0     0
  8| 1  7   103   104   105   106   108   113   114
  9| 1  3   176   177   179     0     0     0     0
*10| 0  2    93   140     0     0     0     0     0   2  17  11
 11| 0  6   145   153   167   175   181   196     0   3  21  15   6   9  19   7
 12| 1  3   126   127   128     0     0     0     0
 13| 1  5    95    96    97    98   100     0     0
 14| 1  4    57    60    61    64     0     0     0
 15| 1  6   154   155   159   161   165   166     0
 16| 1  5    49    50    52    54    55     0     0
 17| 0  4   102   116   125   129     0     0     0  13   8   5  12  20
 18| 1  4    67    73    74    75     0     0     0
 19| 1  7   182   185   186   188   190   192   194
 20| 1  5   131   134   135   136   137     0     0
 21| 1  4   146   149   150   152     0     0     0
 22| 1  4    82    83    86    91     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 195 for incluido, qual linha,coluna na matriz?	Se o 99 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
  +-------------------------------------------------------------------------------
  1| 1  3    32    34    35     0     0     0     0
  2| 0  5    36    45    53    58    63     0     0   1  23   6  14  12  20
  3| 1  3   121   122   125     0     0     0     0
  4| 1  5   167   170   171   173   175     0     0
  5| 1  4    78    82    83    85     0     0     0
  6| 1  5    46    47    48    49    52     0     0
  7| 1  5   195   196   197   199   200     0     0
  8| 1  3   134   135   136     0     0     0     0
  9| 1  4    89    90    91    94     0     0     0
*10| 0  3    77   120   165     0     0     0     0   2  19  11  22
 11| 0  4   126   133   137   146     0     0     0   3  21   8  24  15
 12| 1  3    59    60    61     0     0     0     0
 13| 1  7   183   187   188   189   190   191   193
 14| 1  3    54    55    57     0     0     0     0
 15| 1  7   147   148   153   161   162   163   164
 16| 1  6   111   113   114   116   117   119     0
 17| 1  3   177   178   179     0     0     0     0
 18| 1  4   100   106   108   109     0     0     0
 19| 0  3    86    97   110     0     0     0     0   5   9  18  16
 20| 1  5    65    66    67    74    75     0     0
 21| 1  4   128   129   130   132     0     0     0
 22| 0  3   176   182   194     0     0     0     0   4  17  13   7
 23| 1  4    39    40    41    42     0     0     0
 24| 1  4   138   139   144   145     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 149 for incluido, qual linha,coluna na matriz?	Se o 157 for incluído, qual linha,coluna na matriz?

305-76344 - gar a

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76351 HENRI KIM PINHEIRO RODRIGUES
24CC1305 - 16 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

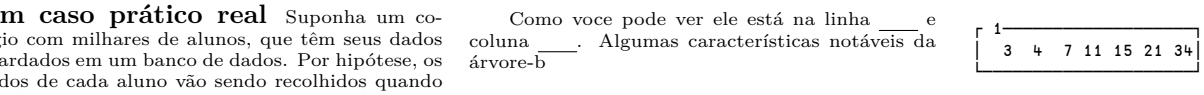
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

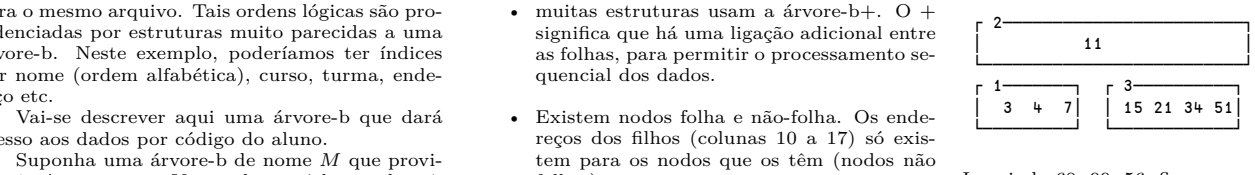
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

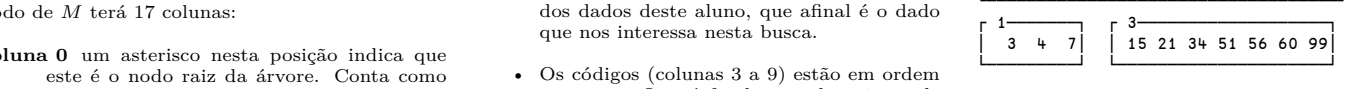


Agora, vamos inserir uma nova chave, o que vai provocar o split

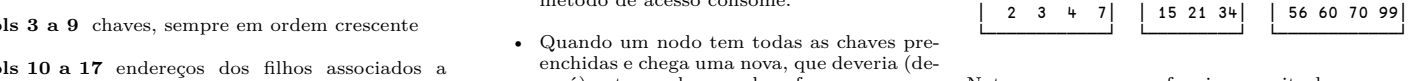
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 3    31   32   33    0    0    0    0
 2| 0 3    34   47   53    0    0    0    0  1 16  9  5
 3| 1 4    77   78   79   80    0    0    0
 4| 1 4   129  130  131  132    0    0    0
 5| 1 3    54   56   57    0    0    0    0
 6| 1 5   174  175  180  182  183    0    0
 7| 1 7   160  163  164  165  166  168  169
 8| 1 5   102  103  104  105  107    0    0
 9| 1 4    48   50   51   52    0    0    0
*10| 0 3    61   98  139    0    0    0    0  2 19 11 22
 11| 0 3   108  127  133    0    0    0    0  8 13  4 18
 12| 1 5    62   63   64   66   67    0    0
 13| 1 7   109  111  112  116  119  120  123
 14| 1 5   140  144  145  147  148    0    0
 15| 1 4    92   93   94   95    0    0    0
 16| 1 7    35   37   38   39   40   41   43
 17| 1 4    85   87   89   90    0    0    0
 18| 1 4   135  136  137  138    0    0    0
 19| 0 4    68   76   84   91    0    0    0  0 12 21  3 17 15
 20| 1 3   154  155  156    0    0    0    0
 21| 1 4    69   71   73   75    0    0    0
 22| 0 4   149  158  172  188    0    0    0  0 14 20  7  6 23
 23| 1 5   189  191  192  198  200    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 170 for incluido, qual linha,coluna na matriz?	Se o 181 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 4    31   32   33   35    0    0    0
 2| 0 3    37   54   62    0    0    0    0  1  8 16  7
 3| 1 4   139  140  141  142    0    0    0
 4| 1 5    81   84   85   87   88    0    0
 5| 1 4   157  159  160  161    0    0    0
 6| 1 6   118  120  121  122  124  125    0
 7| 1 3    63   64   68    0    0    0    0
 8| 1 6    39   42   45   47   51   52    0
 9| 1 7   127  128  129  132  133  135  137
*10| 0 3    69  115  155    0    0    0    0  2 22 11 19
 11| 0 3   126  138  143    0    0    0    0  6  9  3 17
 12| 1 3   178  179  180    0    0    0    0
 13| 1 5   103  104  107  108  109    0    0
 14| 1 7    90   93   94   96   97   99  100
 15| 1 5   189  190  191  194  198    0    0
 16| 1 4    57   58   59   60    0    0    0
 17| 1 7   144  146  147  149  152  153  154
 18| 1 6   164  170  171  173  175  176    0
 19| 0 4   162  177  181  187    0    0    0  5 18 12 21 15
 20| 1 4    71   72   73   75    0    0    0
 21| 1 4   182  183  184  185    0    0    0
 22| 0 3    76   89  101    0    0    0    0 20  4 14 13
```

Responda agora:

Qual a altura da árvore B ?	Se o 48 for incluido, qual linha,coluna na matriz?	Se o 105 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76368 HENRIQUE NOGUEIRA RIBEIRO
24CC1305 - 17 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

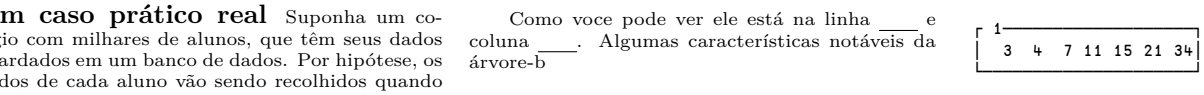
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

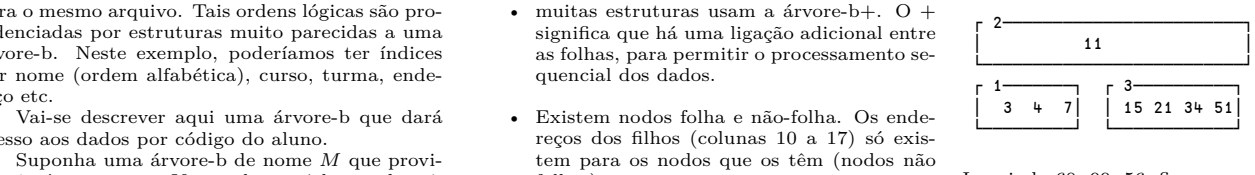
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

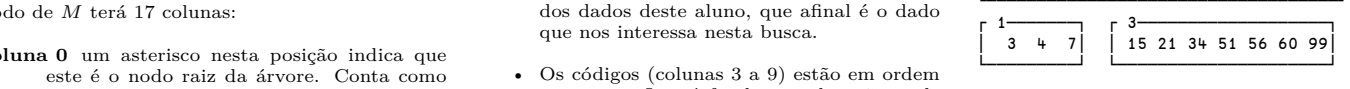


Agora, vamos inserir uma nova chave, o que vai provocar o split

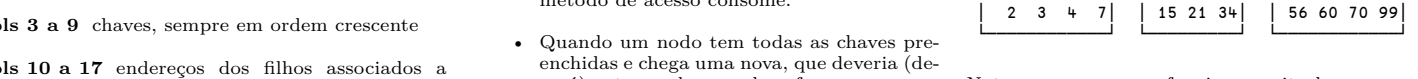
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  3    31    32    36     0     0     0     0
 2 | 0  5    37    48    54    59    70     0     0   1  22  13  21   5   8
 3 | 1  5    98    99   100   101   105     0     0
 4 | 1  7   153   154   155   156   161   166   170
 5 | 1  6    60    61    62    63    65    68     0
 6 | 1  4   130   131   132   133     0     0     0
 7 | 1  3   174   175   176     0     0     0     0
 8 | 1  4    71    73    74    75     0     0     0
 9 | 1  4   135   136   137   138     0     0     0
*10| 0  2    76   129     0     0     0     0     0   2  17  11
11 | 0  6   134   142   151   171   177   190     0   6   9  19   4   7  18  12
12 | 1  6   191   192   193   196   199   200     0
13 | 1  3    50    52    53     0     0     0     0
14 | 1  4   109   110   111   113     0     0     0
15 | 1  5    77    80    81    82    83     0     0
16 | 1  6    87    88    90    91    92    93     0
17 | 0  4    86    96   108   118     0     0     0  15  16   3  14  20
18 | 1  5   179   183   184   187   189     0     0
19 | 1  6   143   144   145   146   147   149     0
20 | 1  4   120   121   125   127     0     0     0
21 | 1  4    55    56    57    58     0     0     0
22 | 1  4    38    39    44    46     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 185 for incluido, qual linha,coluna na matriz?	Se o 141 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  6    31    32    33    35    36    37     0
 2 | 0  3    39    44    59     0     0     0     0   1  15   9  18
 3 | 1  6    96    98   100   103   104   106     0
 4 | 1  3   159   160   161     0     0     0     0
 5 | 1  7    67    68    69    70    71    72    74
 6 | 1  5   188   189   190   192   194     0     0
 7 | 1  6   118   120   124   128   130   131     0
 8 | 1  7   142   145   148   150   153   156   157
 9 | 1  7    45    46    48    51    53    56    58
*10| 0  3    65   117   162     0     0     0     0   2  19  11  21
11 | 0  3   133   141   158     0     0     0     0   7  20   8   4
12 | 1  4   173   174   175   176     0     0     0
13 | 1  7    77    80    82    83    86    89    92
14 | 1  4   110   112   115   116     0     0     0
15 | 1  4    40    41    42    43     0     0     0
16 | 1  5   164   165   167   168   171     0     0
17 | 1  3   197   198   199     0     0     0     0
18 | 1  4    61    62    63    64     0     0     0
19 | 0  3    75    93   107     0     0     0     0   5  13   3  14
20 | 1  3   134   135   140     0     0     0     0
21 | 0  4   172   177   187   195     0     0     0  16  12  22   6  17
22 | 1  3   178   180   184     0     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 84 for incluido, qual linha,coluna na matriz?	Se o 129 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76494 HENRIQUE RANKEL CRENCHIGLOVA
24CC1305 - 18 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

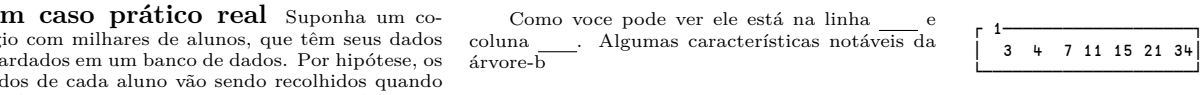
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

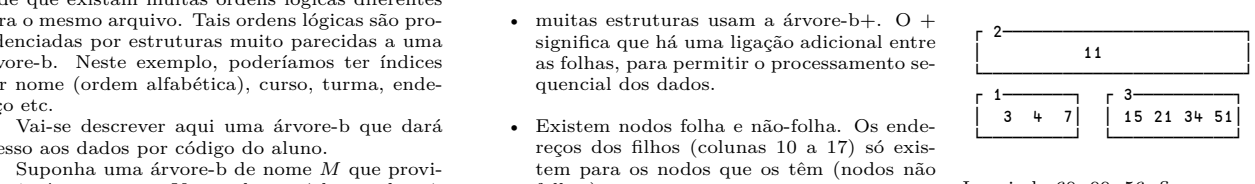
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

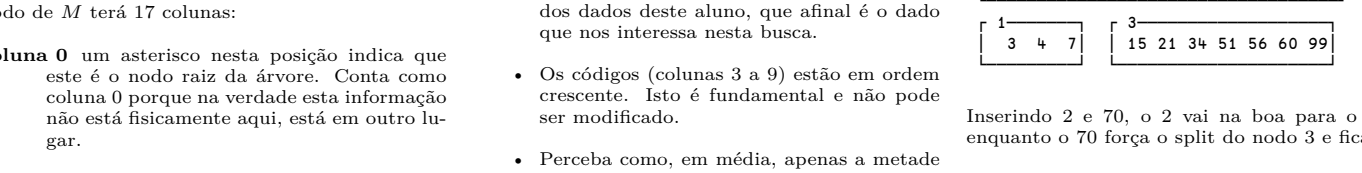


Agora, vamos inserir uma nova chave, o que vai provocar o split

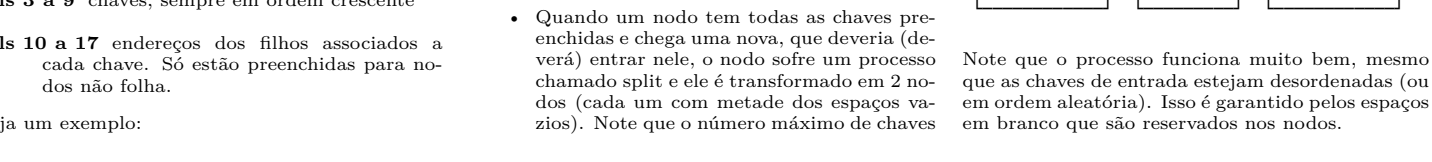
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 7   33  34  35  36  39  40  42
 2| 0 5   45  52  63  73  80   0   0  1 12 20 14 22  6
 3| 1 6  132 134 135 137 140 141   0
 4| 1 3  124 125 127   0   0   0   0
 5| 1 3  155 158 159   0   0   0   0
 6| 1 6   84  88  89  90  92  94   0
 7| 1 7  143 146 147 148 150 151 153
 8| 1 7  179 182 183 185 186 187 188
 9| 1 4   97  98  99 102   0   0   0
*10| 0 2   96 128   0   0   0   0   0  2 18 11
 11| 0 6  142 154 161 172 178 190   0  3  7  5 16 19  8 15
 12| 1 3   46  50  51   0   0   0   0
 13| 1 4  113 114 115 116   0   0   0
 14| 1 4   64  66  69  72   0   0   0
 15| 1 7  192 193 194 195 196 198 199
 16| 1 5  162 163 166 169 171   0   0
 17| 1 4  105 106 107 108   0   0   0
 18| 0 4  104 112 117 122   0   0   0  9 17 13 21  4
 19| 1 3  174 175 176   0   0   0   0
 20| 1 4   54  55  59  62   0   0   0
 21| 1 3  118 119 121   0   0   0   0
 22| 1 3   74  75  76   0   0   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 31 for incluido, qual linha,coluna na matriz?	Se o 144 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3   4   5   6   7   8   9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------
 1| 1 4   31  33  34  36   0   0   0
 2| 0 6   38  46  52  64  77  89   0  1 22  6 21  7 15  9
 3| 1 3  134 135 136   0   0   0   0
 4| 1 7   97  98 100 101 103 104 108
 5| 1 4  178 179 182 185   0   0   0
 6| 1 3   47  49  50   0   0   0   0
 7| 1 5   65  68  69  70  72   0   0
 8| 1 5  149 151 155 158 159   0   0
 9| 1 5   90  91  92  93  94   0   0
*10| 0 2   96 148   0   0   0   0   0  2 11 17
 11| 0 5  110 119 125 132 137   0   0  4 12 23 16  3 18
 12| 1 4  111 112 115 116   0   0   0
 13| 1 4  195 197 199 200   0   0   0
 14| 1 3  162 163 165   0   0   0   0
 15| 1 7   80  81  82  83  84  86  87
 16| 1 3  126 128 130   0   0   0   0
 17| 0 5  161 167 177 186 192   0   0  8 14 20  5 19 13
 18| 1 6  138 139 140 142 145 146   0
 19| 1 3  187 188 191   0   0   0   0
 20| 1 5  170 172 173 174 175   0   0
 21| 1 4   53  55  62  63   0   0   0
 22| 1 3   39  40  44   0   0   0   0
 23| 1 4  120 121 122 124   0   0   0
```

Responda agora:

Qual a altura da árvore B ?	Se o 113 for incluido, qual linha,coluna na matriz?	Se o 157 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76375 JOAO FRANCISCO LEONEL DE SOUSA
24CC1305 - 19 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

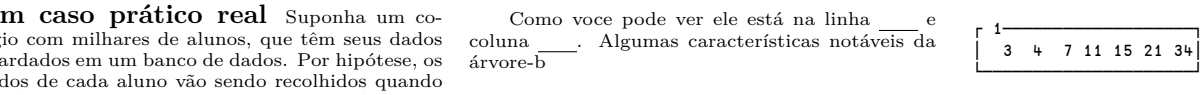
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

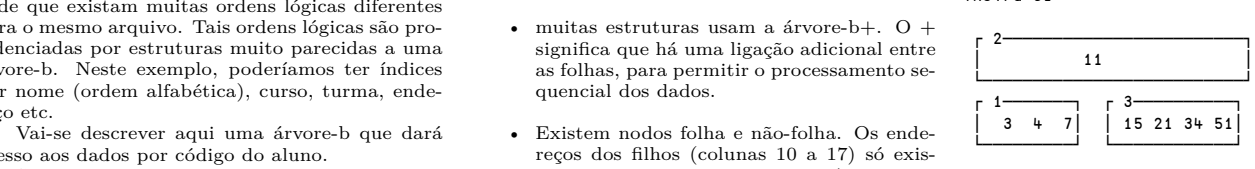
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

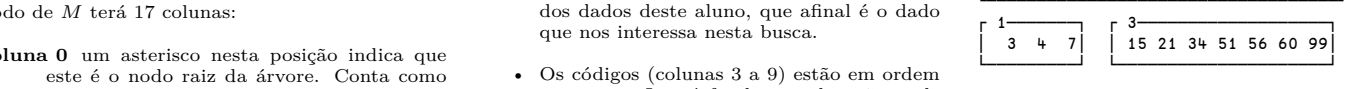


Agora, vamos inserir uma nova chave, o que vai provocar o split

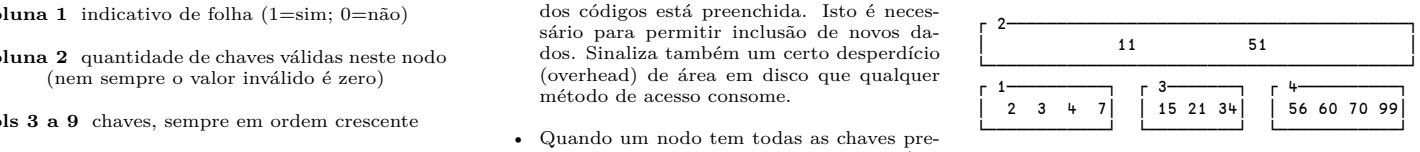
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  4    31    33    36    37     0     0     0
 2 | 0  4    39    48    57    65     0     0     0   1  12  17   5  20
 3 | 1  7    90    91    94    95    97    98    99
 4 | 1  7   140   142   144   147   148   149   150
 5 | 1  5    58    59    60    61    63     0     0
 6 | 1  6   179   180   182   183   184   185     0
 7 | 1  7   115   117   119   120   122   123   124
 8 | 1  5    72    73    74    78    79     0     0
 9 | 1  6   167   170   173   175   176   177     0
*10| 0  2    70   114     0     0     0     0     0   2  19  11
11 | 0  6   127   139   152   166   178   191     0   7  13   4  14   9   6  15
12 | 1  5    40    41    42    45    46     0     0
13 | 1  5   129   130   135   136   138     0     0
14 | 1  4   153   162   164   165     0     0     0
15 | 1  5   192   195   196   197   199     0     0
16 | 1  6   101   105   107   110   111   112     0
17 | 1  4    51    52    53    55     0     0     0
18 | 1  6    82    83    84    85    86    88     0
19 | 0  3    81    89   100     0     0     0     0   8  18   3  16
20 | 1  3    66    67    69     0     0     0     0
```

Responda agora:

Qual a altura da árvore B ?	Se o 43 for incluido, qual linha,coluna na matriz?	Se o 188 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2     3     4     5     6     7     8     9  10  11  12  13  14  15  16  17
   +-------------------------------------------------------------------------------
 1 | 1  6    32    33    38    39    40    41     0
 2 | 0  4    42    52    58    68     0     0     0   1  14  16  20   6
 3 | 1  4   135   137   138   139     0     0     0
 4 | 1  4    89    90    91    94     0     0     0
 5 | 1  6   127   128   129   130   131   133     0
 6 | 1  7    70    71    74    75    77    78    80
 7 | 1  7   160   163   165   166   169   170   172
 8 | 1  4   175   176   177   180     0     0     0
 9 | 1  4   101   103   104   108     0     0     0
*10| 0  3    81   126   158     0     0     0     0   2  19  11  22
11 | 0  3   134   140   148     0     0     0     0   5   3  21  15
12 | 1  6    82    83    84    85    86    87     0
13 | 1  4   182   185   186   187     0     0     0
14 | 1  4    43    45    47    49     0     0     0
15 | 1  4   151   152   155   156     0     0     0
16 | 1  3    54    55    56     0     0     0     0
17 | 1  6   113   114   115   120   121   124     0
18 | 1  5   192   195   197   198   200     0     0
19 | 0  3    88    99   112     0     0     0     0  12   4   9  17
20 | 1  5    59    64    65    66    67     0     0
21 | 1  5   142   143   144   146   147     0     0
22 | 0  3   174   181   188     0     0     0     0   7   8  13  18
```

Responda agora:

Qual a altura da árvore B ?	Se o 196 for incluido, qual linha,coluna na matriz?	Se o 119 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76382 LANA ALESSANDRA GALEGO VEIGA 24CC1305 - 20 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

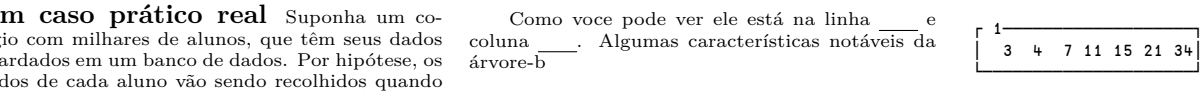
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

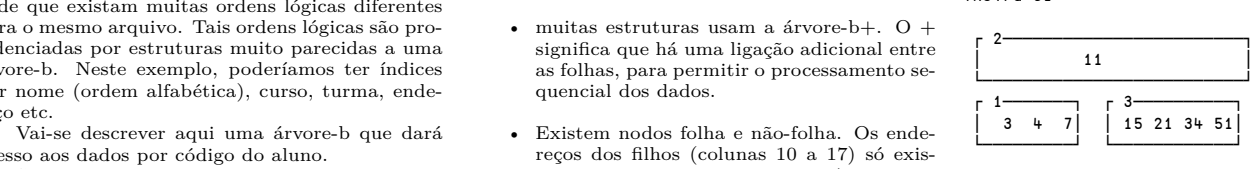
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

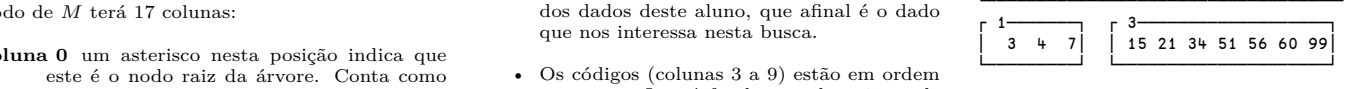


Agora, vamos inserir uma nova chave, o que vai provocar o split

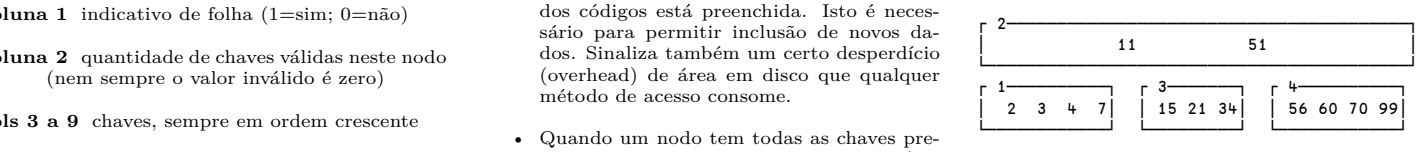
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 4    31   32   34   35    0    0    0
 2| 0 5    36   41   49   64   72    0    0   1 23 13 12  4 25
 3| 1 5   123  125  127  129  132    0    0
 4| 1 3    65   69   70    0    0    0    0
 5| 1 4   168  170  171  172    0    0    0
 6| 1 6    83   84   85   87   90   91    0
 7| 1 3   103  104  105    0    0    0    0
 8| 1 5   185  187  189  191  192    0    0
 9| 1 3   152  154  155    0    0    0    0
*10| 0 3    82  121  166    0    0    0    0  2 20 11 21
 11| 0 4   133  141  150  156    0    0    0  3 16 18  9 24
 12| 1 6    53   55   56   57   61   62    0
 13| 1 3    44   45   47    0    0    0    0
 14| 1 4   175  176  178  180    0    0    0
 15| 1 4    96   97   99  100    0    0    0
 16| 1 4   134  135  138  140    0    0    0
 17| 1 4   194  198  199  200    0    0    0
 18| 1 4   142  145  147  148    0    0    0
 19| 1 3   108  110  112    0    0    0    0
 20| 0 4    93  101  107  115    0    0    0  6 15  7 19 22
 21| 0 3   173  182  193    0    0    0    0  5 14  8 17
 22| 1 4   116  117  119  120    0    0    0
 23| 1 3    38   39   40    0    0    0    0
 24| 1 4   157  158  159  163    0    0    0
 25| 1 5    75   77   78   79   81    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 195 for incluido, qual linha,coluna na matriz?	Se o 164 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5    31   32   35   37   38    0    0
 2| 0 6    41   53   61   72   89   95    0   1 14  7 16  5  8 15
 3| 1 3   102  103  106    0    0    0    0
 4| 1 3   149  150  151    0    0    0    0
 5| 1 7    75   80   81   82   83   86   87
 6| 1 4   183  185  189  191    0    0    0
 7| 1 3    56   59   60    0    0    0    0
 8| 1 4    90   91   92   94    0    0    0
 9| 1 7   119  120  121  122  123  124  125
*10| 0 2   101  152    0    0    0    0    0  2 11 19
 11| 0 4   109  117  130  147    0    0    0  3 21  9 17  4
 12| 1 7   159  160  161  162  163  165  166
 13| 1 7   169  170  171  174  176  177  179
 14| 1 7    42   46   47   48   49   51   52
 15| 1 4    96   97   99  100    0    0    0
 16| 1 4    62   64   66   68    0    0    0
 17| 1 6   131  135  136  138  142  146    0
 18| 1 4   154  155  156  157    0    0    0
 19| 0 4   158  168  182  193    0    0    0  18 12 13  6 20
 20| 1 4   194  196  199  200    0    0    0
 21| 1 5   110  113  114  115  116    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 40 for incluido, qual linha,coluna na matriz?	Se o 58 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76399 LORENZO QUIRINO DE SALLES
24CC1305 - 21 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

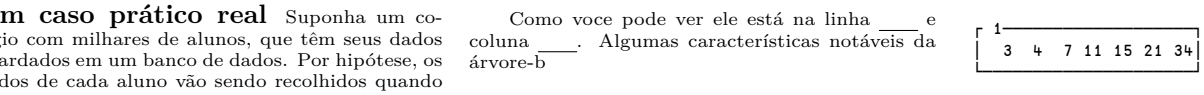
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

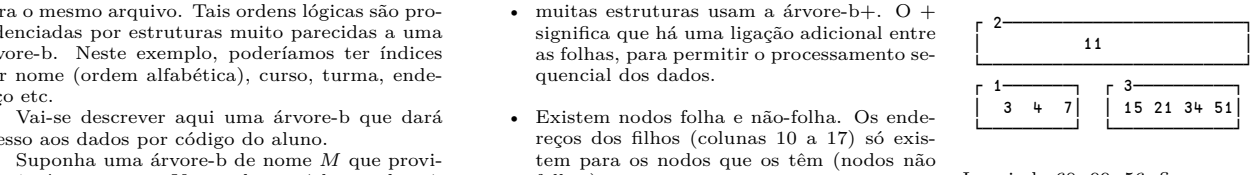
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

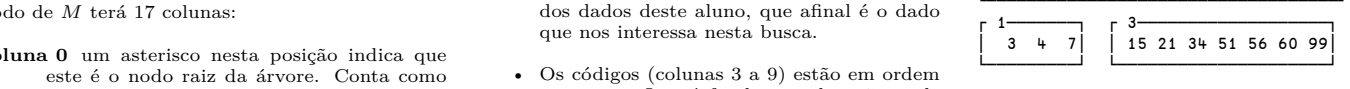


Agora, vamos inserir uma nova chave, o que vai provocar o split

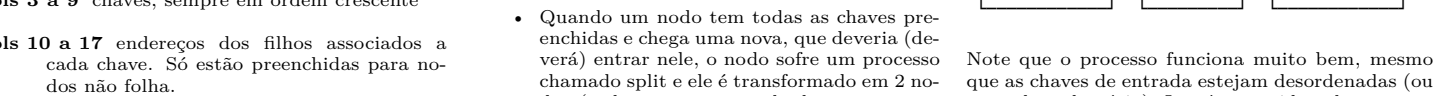
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 3    32   33   34    0    0    0    0
 2| 0 3    35   49   59    0    0    0    0  1 22  7  9
 3| 1 5    91   93   94   95   96    0    0
 4| 1 4   148  149  150  152    0    0    0
 5| 1 6   124  126  128  130  132  133    0
 6| 1 3   171  173  174    0    0    0    0
 7| 1 5    50   51   52   54   56    0    0
 8| 1 3   108  109  110    0    0    0    0
 9| 1 4    60   62   63   65    0    0    0
*10| 0 3    68  106  147    0    0    0    0  2 23 11 17
 11| 0 3   111  123  135    0    0    0    0  8 14  5 16
 12| 1 4   189  190  191  193    0    0    0
 13| 1 4    69   70   72   73    0    0    0
 14| 1 7   112  113  115  117  118  121  122
 15| 1 6   156  157  159  164  165  167    0
 16| 1 5   137  139  142  143  146    0    0
 17| 0 5   153  169  176  188  194    0    0  4 15  6 21 12 18
 18| 1 4   195  196  197  200    0    0    0
 19| 1 3    99  102  103    0    0    0    0
 20| 1 6    76   77   79   82   83   85    0
 21| 1 6   177  178  181  184  185  187    0
 22| 1 5    39   41   42   43   44    0    0
 23| 0 3    75   87   97    0    0    0    0 13 20  3 19
```

Responda agora:

Qual a altura da árvore B ?	Se o 78 for incluido, qual linha,coluna na matriz?	Se o 98 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 4    34   35   37   40    0    0    0
 2| 0 4    43   51   62   73    0    0    0  1 14 13  4 19
 3| 1 6   123  125  127  128  129  131    0
 4| 1 7    63   64   66   68   69   70   72
 5| 1 4   160  161  163  164    0    0    0
 6| 1 4   148  149  151  152    0    0    0
 7| 1 7   133  136  137  140  141  142  144
 8| 1 4   196  197  198  200    0    0    0
 9| 1 4    90   99  101  105    0    0    0
*10| 0 2    79  132    0    0    0    0    0  2 18 11
 11| 0 7   146  153  159  167  176  185  193  7  6 22  5 12 21 16  8
 12| 1 4   168  170  174  175    0    0    0
 13| 1 7    52   53   55   56   57   60   61
 14| 1 5    44   47   48   49   50    0    0
 15| 1 5    80   82   84   85   87    0    0
 16| 1 5   186  189  190  191  192    0    0
 17| 1 4   110  111  112  114    0    0    0
 18| 0 4    89  107  115  122    0    0    0  0 15  9 17 20  3
 19| 1 3    75   76   78    0    0    0    0
 20| 1 3   116  117  120    0    0    0    0
 21| 1 4   179  180  183  184    0    0    0
 22| 1 3   154  155  157    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 33 for incluido, qual linha,coluna na matriz?	Se o 81 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76401 LUAN TEIXEIRA DIBE
24CC1305 - 22 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

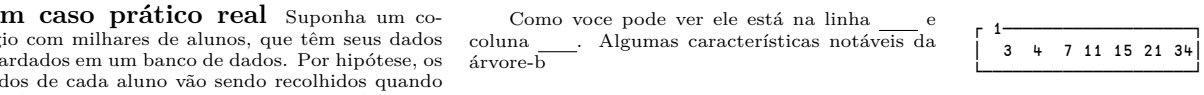
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

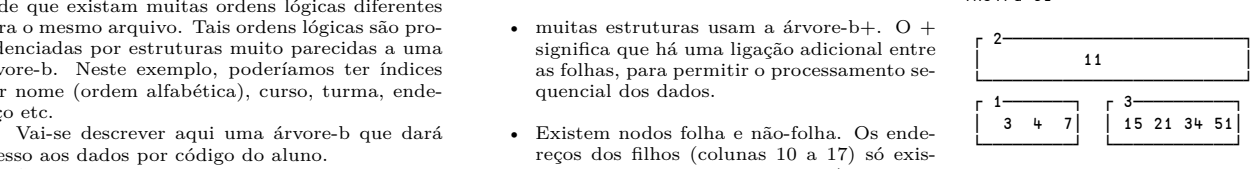
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

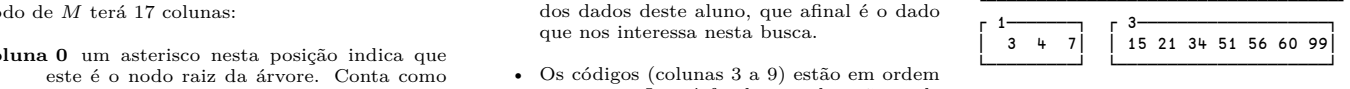


Agora, vamos inserir uma nova chave, o que vai provocar o split

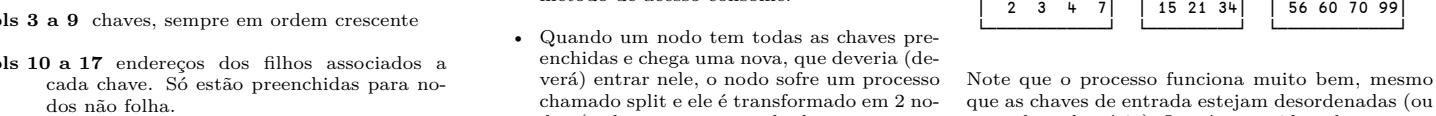
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5    32   34   35   36   41    0    0
 2| 0 3    43   48   56    0    0    0    0  1 18 16  8
 3| 1 5    87   89   90   91   92    0    0
 4| 1 3   165  166  168    0    0    0    0
 5| 1 7   177  178  180  181  182  183  184
 6| 1 3   127  130  134    0    0    0    0
 7| 1 6    68   70   72   74   75   76    0
 8| 1 6    57   59   60   61   63   64    0
 9| 1 7   110  114  115  116  117  120  123
*10| 0 3    66  104  152    0    0    0    0  2 21 11 20
 11| 0 3   124  135  145    0    0    0    0  9  6 19 13
 12| 1 3   190  191  192    0    0    0    0
 13| 1 5   147  148  149  150  151    0    0
 14| 1 4    99  100  101  102    0    0    0
 15| 1 6   153  154  156  158  159  160    0
 16| 1 4    49   50   51   55    0    0    0
 17| 1 3    82   83   84    0    0    0    0
 18| 1 4    44   45   46   47    0    0    0
 19| 1 4   136  138  142  143    0    0    0
 20| 0 5   162  169  176  187  194    0    0  0 15  4 23  5 12 22
 21| 0 3    78   85   98    0    0    0    0  7 17  3 14
 22| 1 4   195  196  197  199    0    0    0
 23| 1 4   170  172  173  174    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 175 for incluido, qual linha,coluna na matriz?	Se o 42 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 6    32   38   39   40   41   42    0
 2| 0 3    43   63   74    0    0    0    0  1 14  9 16
 3| 1 7   122  123  124  128  129  130  131
 4| 1 7   169  171  172  176  177  180  181
 5| 1 7    95   96   98   99  100  103  105
 6| 1 5   139  143  144  145  146    0    0
 7| 1 3    82   83   84    0    0    0    0
 8| 1 4   133  134  135  136    0    0    0
 9| 1 5    64   65   66   67   71    0    0
*10| 0 2    81  121    0    0    0    0    0  2 19 11
 11| 0 7   132  137  147  161  168  183  189  3  8  6 15 17  4 13 21
 12| 1 6   107  108  109  110  111  115    0
 13| 1 3   184  185  187    0    0    0    0
 14| 1 7    45   47   48   51   53   61   62
 15| 1 5   154  155  158  159  160    0    0
 16| 1 4    75   77   79   80    0    0    0
 17| 1 3   163  165  166    0    0    0    0
 18| 1 3   117  118  120    0    0    0    0
 19| 0 4    85   94  106  116    0    0    0  7 20  5 12 18
 20| 1 5    86   88   89   90   92    0    0
 21| 1 4   190  191  198  200    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 54 for incluido, qual linha,coluna na matriz?	Se o 36 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76418 MURILO DIRCKSEN CAMPOS
24CC1305 - 23 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

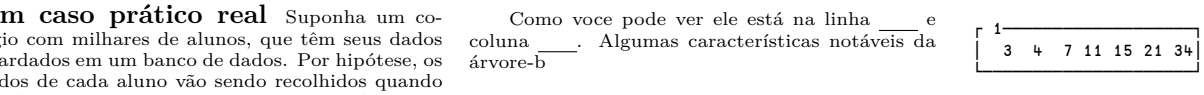
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

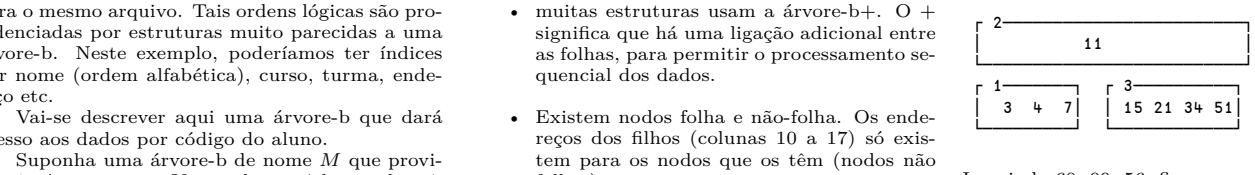
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

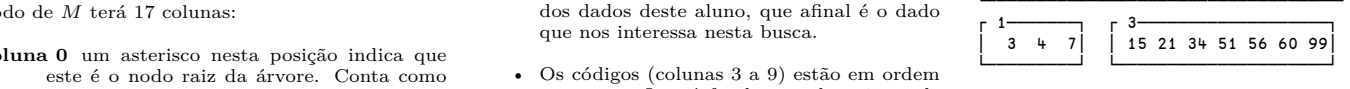


Agora, vamos inserir uma nova chave, o que vai provocar o split

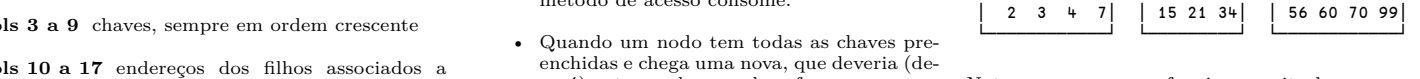
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1 | 1 4    31   33   34   35    0    0    0
 2 | 0 5    36   52   63   78   88    0    0  1 15  9  4 16  7
 3 | 1 7    99  102  104  107  108  109  110
 4 | 1 7    65   68   69   71   72   73   76
 5 | 1 6   152  154  157  158  159  164    0
 6 | 1 5   134  135  136  138  139    0    0
 7 | 1 5    89   90   94   96   97    0    0
 8 | 1 6   181  183  184  185  187  188    0
 9 | 1 7    54   56   57   58   59   61   62
*10| 0 2    98  165    0    0    0    0    0  2 11 18
 11| 0 5   111  128  133  142  151    0    0  3 12 20  6 19  5
 12| 1 4   113  115  123  125    0    0    0
 13| 1 6   166  168  170  171  172  173    0
 14| 1 6   191  192  193  195  198  200    0
 15| 1 4    43   46   48   49    0    0    0
 16| 1 6    79   82   83   84   85   87    0
 17| 1 3   175  177  179    0    0    0    0
 18| 0 3   174  180  190    0    0    0    0 13 17  8 14
 19| 1 6   143  144  146  147  148  150    0
 20| 1 3   129  130  131    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 67 for incluido, qual linha,coluna na matriz?	Se o 161 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1 | 1 5    33   34   36   37   38    0    0
 2 | 0 6    39   46   54   62   71   81    0  1 12 20  4  8 19  3
 3 | 1 5    84   85   86   88   89    0    0
 4 | 1 5    55   57   58   60   61    0    0
 5 | 1 3   135  137  138    0    0    0    0
 6 | 1 5   116  117  119  120  121    0    0
 7 | 1 3   189  190  191    0    0    0    0
 8 | 1 4    66   68   69   70    0    0    0
 9 | 1 5   100  102  103  104  105    0    0
*10| 0 3    98  132  172    0    0    0    0  2 11 17 24
 11| 0 3   107  115  124    0    0    0    0  9 15  6 14
 12| 1 4    41   42   43   44    0    0    0
 13| 1 4   153  154  157  160    0    0    0
 14| 1 4   125  126  128  131    0    0    0
 15| 1 5   108  109  111  113  114    0    0
 16| 1 4   193  194  195  196    0    0    0
 17| 0 3   139  151  161    0    0    0    0  5 22 13 18
 18| 1 4   162  163  166  169    0    0    0
 19| 1 6    73   74   76   77   79   80    0
 20| 1 5    47   48   50   51   53    0    0
 21| 1 4   178  179  180  182    0    0    0
 22| 1 4   143  144  147  148    0    0    0
 23| 1 3   198  199  200    0    0    0    0
 24| 0 3   185  192  197    0    0    0    0 21  7 16 23
```

Responda agora:

Qual a altura da árvore B ?	Se o 158 for incluido, qual linha,coluna na matriz?	Se o 82 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76425 OLAVO AUGUSTO ROSSI
24CC1305 - 24 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

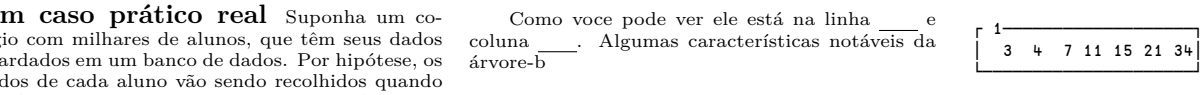
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

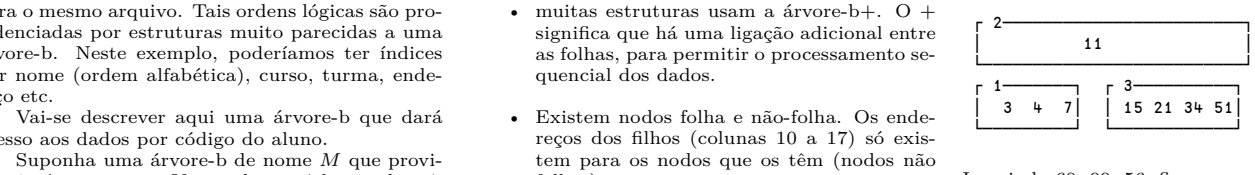
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

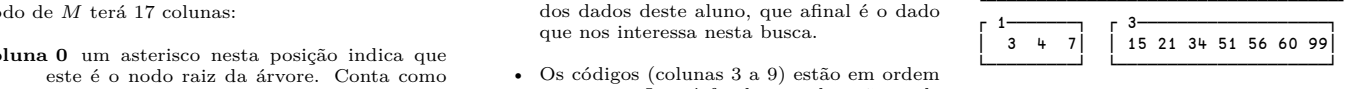


Agora, vamos inserir uma nova chave, o que vai provocar o split

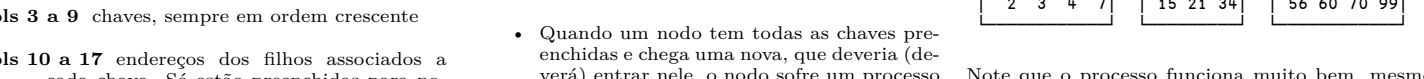
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 7    31   33   34   35   38   41   42
 2| 0 3    45   53   63    0    0    0    0  1 17  5 15
 3| 1 4   148  149  150  151    0    0    0
 4| 1 3   108  111  113    0    0    0    0
 5| 1 6    54   56   57   59   60   62    0
 6| 1 5    71   72   74   77   78    0    0
 7| 1 7   165  166  167  169  170  173  174
 8| 1 3   124  125  126    0    0    0    0
 9| 1 3   186  188  189    0    0    0    0
*10| 0 3    70  122  153    0    0    0    0  2 19 11 22
 11| 0 3   129  138  147    0    0    0    0  8 14 21  3
 12| 1 6    92   93   98   99  103  105    0
 13| 1 5   157  159  160  162  163    0    0
 14| 1 4   130  131  132  135    0    0    0
 15| 1 3    64   66   67    0    0    0    0
 16| 1 5   176  178  179  181  182    0    0
 17| 1 3    46   48   52    0    0    0    0
 18| 1 6    80   81   83   84   85   87    0
 19| 0 4    79   89  107  114    0    0    0  6 18 12  4 20
 20| 1 5   115  117  118  120  121    0    0
 21| 1 4   139  141  143  145    0    0    0
 22| 0 4   164  175  184  190    0    0    0 13  7 16  9 23
 23| 1 4   195  196  197  200    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 199 for incluido, qual linha,coluna na matriz?	Se o 82 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 3    31   32   35    0    0    0    0
 2| 0 6    39   50   58   68   74   81    0  1 21 13 19  5 22 15
 3| 1 6    93   94   97   98  100  101    0
 4| 1 4   151  152  153  154    0    0    0
 5| 1 4    69   71   72   73    0    0    0
 6| 1 5   134  135  137  138  141    0    0
 7| 1 6   163  164  165  166  168  171    0
 8| 1 4   112  115  117  119    0    0    0
 9| 1 4   144  145  148  149    0    0    0
*10| 0 2    91  132    0    0    0    0    0  2 18 11
 11| 0 7   142  150  155  162  174  181  192  6  9  4 23  7 12 20 16
 12| 1 4   175  176  177  178    0    0    0
 13| 1 5    51   52   53   55   56    0    0
 14| 1 5   104  105  107  108  109    0    0
 15| 1 4    84   86   87   88    0    0    0
 16| 1 5   193  194  195  199  200    0    0
 17| 1 3   123  128  131    0    0    0    0
 18| 0 3   103  110  120    0    0    0    0  3 14  8 17
 19| 1 5    60   61   65   66   67    0    0
 20| 1 4   182  183  188  190    0    0    0
 21| 1 4    43   44   46   49    0    0    0
 22| 1 4    75   77   79   80    0    0    0
 23| 1 3   156  158  160    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 41 for incluido, qual linha,coluna na matriz?	Se o 189 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76432 PAULO GUSTAVO TORRES
24CC1305 - 25 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

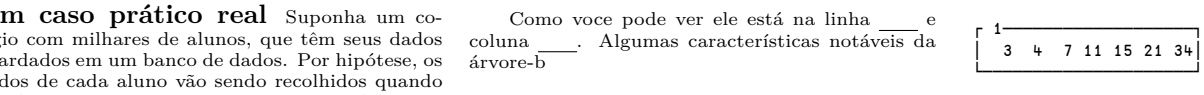
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

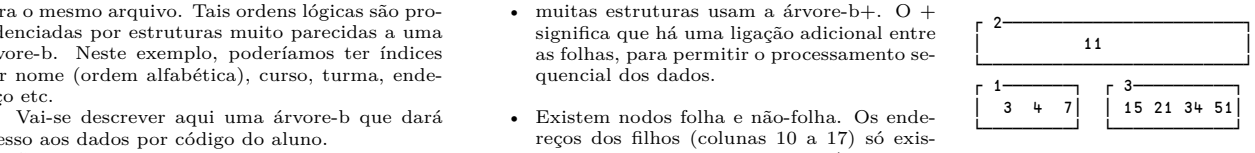
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

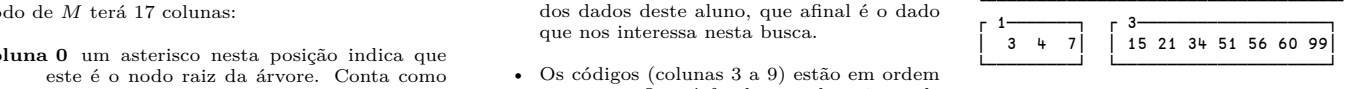


Agora, vamos inserir uma nova chave, o que vai provocar o split

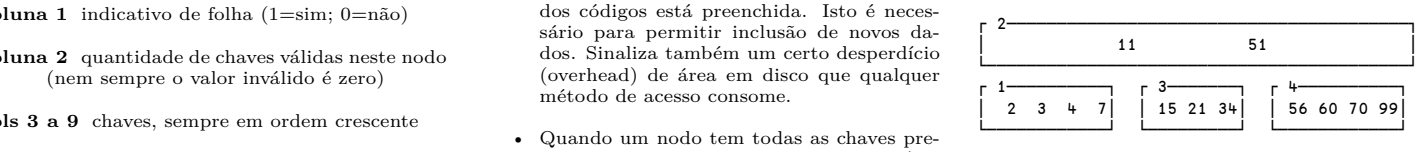
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 3    31   32   33    0    0    0    0
 2| 0 3    34   43   56    0    0    0    0  1 20   6 12
 3| 1 6   129  130  132  135  136  138    0
 4| 1 4    82   84   85   87    0    0    0
 5| 1 7   104  105  106  107  108  109  110
 6| 1 7    44   48   49   50   51   53   55
 7| 1 3   158  161  162    0    0    0    0
 8| 1 5   150  151  153  154  156    0    0
 9| 1 3    91   92   93    0    0    0    0
*10| 0 3    70  102  149    0    0    0    0  2 22 11 18
 11| 0 3   116  128  140    0    0    0    0  5 13   3 17
 12| 1 7    57   59   62   63   66   68   69
 13| 1 5   117  118  119  122  127    0    0
 14| 1 3   191  192  193    0    0    0    0
 15| 1 6   178  179  182  183  185  187    0
 16| 1 3    71   74   76    0    0    0    0
 17| 1 4   142  143  147  148    0    0    0
 18| 0 5   157  164  171  190  195    0    0  8  7 19 15 14 23
 19| 1 5   166  167  168  169  170    0    0
 20| 1 4    35   37   40   41    0    0    0
 21| 1 4    95   96   97  101    0    0    0
 22| 0 3    81   89   94    0    0    0    0 16  4  9 21
 23| 1 4   196  197  199  200    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 198 for incluido, qual linha,coluna na matriz?	Se o 60 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5    35   36   38   40   42    0    0
 2| 0 3    43   47   56    0    0    0    0  1 15   6 14
 3| 1 5    98   99  100  102  103    0    0
 4| 1 4   155  158  159  160    0    0    0
 5| 1 3    84   85   88    0    0    0    0
 6| 1 5    48   49   50   51   52    0    0
 7| 1 4   172  173  175  176    0    0    0
 8| 1 5   121  124  126  129  130    0    0
 9| 1 3    66   68   70    0    0    0    0
*10| 0 3    64   97  142    0    0    0    0  2 22 11 20
 11| 0 4   104  110  120  133    0    0    0  3 13 19   8 25
 12| 1 4   184  187  188  189    0    0    0
 13| 1 4   105  107  108  109    0    0    0
 14| 1 5    57   59   61   62   63    0    0
 15| 1 3    44   45   46    0    0    0    0
 16| 1 5    90   91   94   95   96    0    0
 17| 1 5   143  144  145  148  150    0    0
 18| 1 4    72   77   79   81    0    0    0
 19| 1 4   111  113  116  117    0    0    0
 20| 0 6   153  161  170  177  183  196    0  17   4  21   7  24  12  23
 21| 1 3   166  167  168    0    0    0    0
 22| 0 3    71   82   89    0    0    0    0  9 18   5 16
 23| 1 3   197  198  199    0    0    0    0
 24| 1 4   178  179  180  182    0    0    0
 25| 1 3   134  139  140    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 132 for incluido, qual linha,coluna na matriz?	Se o 122 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76449 PEDRO CAPOTE DA CONCEICAO SOAR
24CC1305 - 26 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ___ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

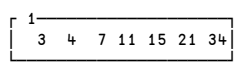
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

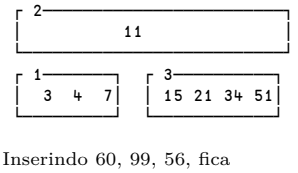
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

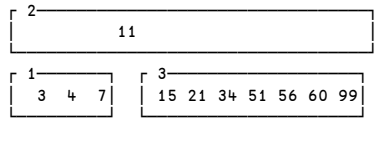


Agora, vamos inserir uma nova chave, o que vai provocar o split

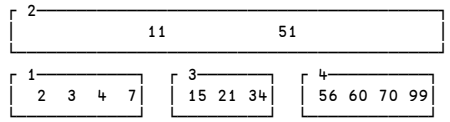
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   31   32   34   36   37    0    0
 2| 0  4   40   50   61   67    0    0    0  1 16  6 21  3
 3| 1  6   68   69   70   72   73   74    0
 4| 1  7  148  149  150  152  154  156  157
 5| 1  3  185  187  188    0    0    0    0
 6| 1  4   52   54   59   60    0    0    0
 7| 1  5  132  138  140  141  143    0    0
 8| 1  6  166  167  169  170  171  172    0
 9| 1  4  101  103  104  105    0    0    0
*10| 0  2   77  130    0    0    0    0    0  2 18 11
11| 0  6  147  159  165  175  184  189    0  7  4 19  8 15  5 13
12| 1  5  114  115  116  120  121    0    0
13| 1  5  192  193  194  196  199    0    0
14| 1  5   78   79   80   82   84    0    0
15| 1  5  176  177  179  180  181    0    0
16| 1  4   42   43   47   48    0    0    0
17| 1  4  123  127  128  129    0    0    0
18| 0  5   86   96  106  113  122    0    0  0 14 22  9 20 12 17
19| 1  3  160  162  163    0    0    0    0
20| 1  3  108  110  112    0    0    0    0
21| 1  3   63   64   65    0    0    0    0
22| 1  6   87   88   90   93   94   95    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 182 for incluido, qual linha,coluna na matriz?	Se o 168 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1  2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1  5   31   34   35   37   38    0    0
 2| 0  3   41   52   65    0    0    0    0  1 16  4 14
 3| 1  6  160  161  162  164  165  166    0
 4| 1  5   54   56   57   60   64    0    0
 5| 1  3  182  183  186    0    0    0    0
 6| 1  5  105  106  107  108  112    0    0
 7| 1  7  122  123  124  125  126  127  128
 8| 1  4  169  170  171  175    0    0    0
 9| 1  6   73   74   75   77   78   80    0
*10| 0  3   72  120  168    0    0    0    0  2 18 11 22
11| 0  3  129  147  158    0    0    0    0  7 13 20  3
12| 1  4  195  196  197  200    0    0    0
13| 1  5  134  137  139  144  146    0    0
14| 1  4   66   67   68   71    0    0    0
15| 1  6   83   84   87   88   89   90    0
16| 1  4   43   44   45   46    0    0    0
17| 1  6   95   97   99  101  102  103    0
18| 0  4   81   94  104  113    0    0    0  9 15 17  6 19
19| 1  3  114  115  118    0    0    0    0
20| 1  3  152  153  157    0    0    0    0
21| 1  3  177  179  180    0    0    0    0
22| 0  4  176  181  187  194    0    0    0  8 21  5 23 12
23| 1  4  188  189  191  192    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 155 for incluido, qual linha,coluna na matriz?	Se o 150 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76456 PEDRO HENRIQUE DE SOUZA
24CC1305 - 27 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

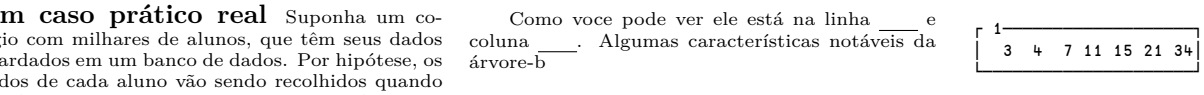
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

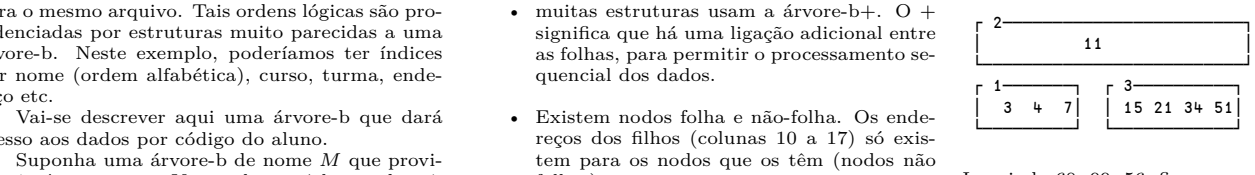
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

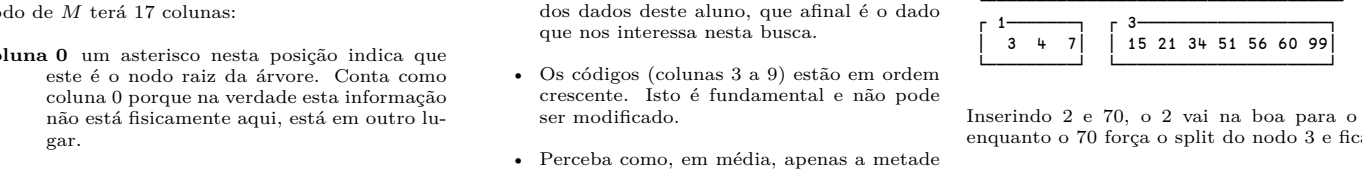


Agora, vamos inserir uma nova chave, o que vai provocar o split

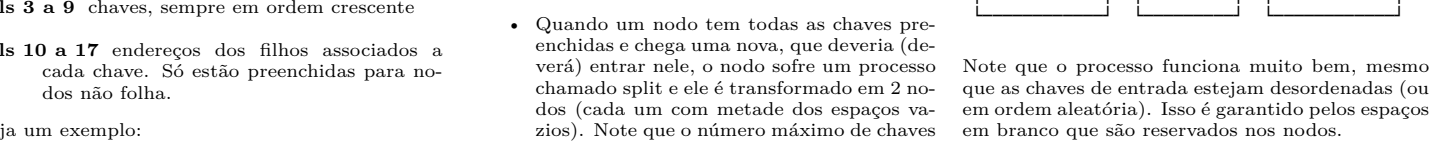
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 4    32   33   34   35    0    0    0
 2| 0 4    36   50   61   74    0    0    0  1 15   8 22   9
 3| 1 5   165  168  169  170  171    0    0
 4| 1 4   142  144  146  147    0    0    0
 5| 1 4   124  125  126  127    0    0    0
 6| 1 6    94   96   97   98   99  100    0
 7| 1 3   183  184  185    0    0    0    0
 8| 1 5    51   54   56   57   58    0    0
 9| 1 3    77   79   80    0    0    0    0
*10| 0 3    82  123  164    0    0    0    0  2 21 11 19
 11| 0 4   128  141  150  158    0    0    0  5 16   4 14 23
 12| 1 5   117  119  120  121  122    0    0
 13| 1 7   102  106  107  108  109  110  112
 14| 1 4   151  152  155  157    0    0    0
 15| 1 6    37   38   42   43   45   48    0
 16| 1 5   131  133  135  137  140    0    0
 17| 1 5   191  192  196  198  200    0    0
 18| 1 4   176  179  180  181    0    0    0
 19| 0 3   174  182  187    0    0    0    0  3 18   7 17
 20| 1 5    83   85   86   88   89    0    0
 21| 0 3    91  101  114    0    0    0    0 20   6 13 12
 22| 1 5    68   69   70   72   73    0    0
 23| 1 3   159  160  161    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 62 for incluido, qual linha,coluna na matriz?	Se o 156 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 3    32   35   36    0    0    0    0
 2| 0 4    42   53   61   76    0    0    0  1 16   4 17 22
 3| 1 5    96  100  102  103  104    0    0
 4| 1 5    54   57   58   59   60    0    0
 5| 1 4   149  151  153  154    0    0    0
 6| 1 4   180  184  186  188    0    0    0
 7| 1 6   131  132  133  134  136  139    0
 8| 1 6   142  143  144  145  146  147    0
 9| 1 4   107  108  111  112    0    0    0
*10| 0 3    81  129  162    0    0    0    0  2 18 11 21
 11| 0 3   140  148  155    0    0    0    0  7  8   5 14
 12| 1 6    82   83   86   87   91   92    0
 13| 1 5   164  165  166  167  169    0    0
 14| 1 4   156  158  159  161    0    0    0
 15| 1 4   115  116  117  118    0    0    0
 16| 1 6    46   47   48   49   51   52    0
 17| 1 5    64   65   68   74   75    0    0
 18| 0 4    93  105  114  121    0    0    0  0 12   3   9 15 23
 19| 1 6   192  194  195  196  199  200    0
 20| 1 4   171  173  175  178    0    0    0
 21| 0 3   170  179  190    0    0    0    0 13 20   6 19
 22| 1 3    77   78   80    0    0    0    0
 23| 1 3   122  123  125    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 62 for incluido, qual linha,coluna na matriz?	Se o 99 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76463 PEDRO HENRIQUE STAATS
24CC1305 - 28 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

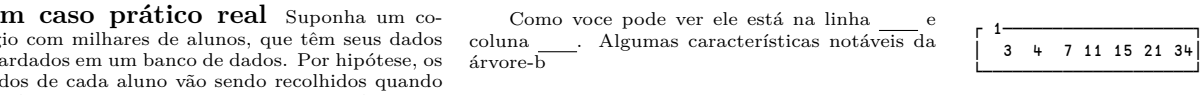
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

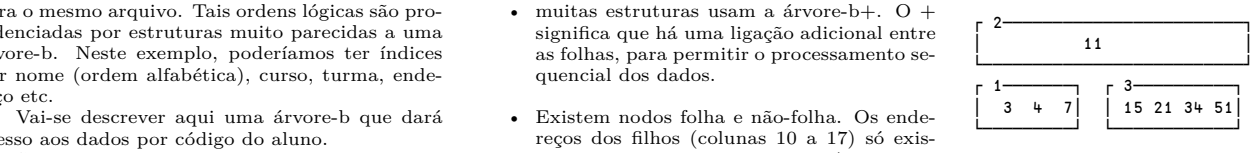
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

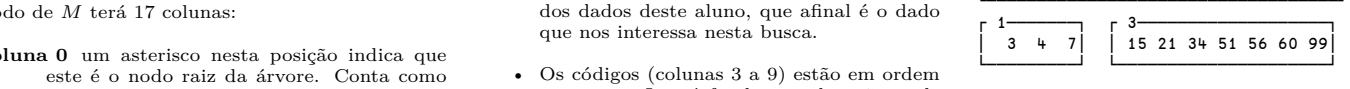


Agora, vamos inserir uma nova chave, o que vai provocar o split

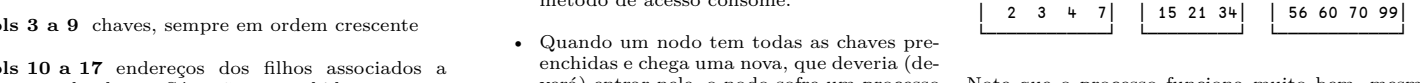
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 5    31   32   33   34   35    0    0
 2| 0 4    36   42   55   65    0    0    0   1 12 23   9 17
 3| 1 7   124  126  127  128  129  130  132
 4| 1 7    87   93   94   96   97   99  100
 5| 1 3   174  175  176    0    0    0    0
 6| 1 3   108  110  111    0    0    0    0
 7| 1 5   192  193  194  195  200    0    0
 8| 1 7   145  147  148  149  150  152  153
 9| 1 3    56   57   61    0    0    0    0
*10| 0 2    72  121    0    0    0    0    0   2 18 11
 11| 0 7   134  142  155  164  173  177  191   3 15  8 14 21  5 19  7
 12| 1 3    38   40   41    0    0    0    0
 13| 1 4    74   75   76   77    0    0    0
 14| 1 4   156  157  158  160    0    0    0
 15| 1 4   136  138  139  141    0    0    0
 16| 1 3   102  105  106    0    0    0    0
 17| 1 4    67   68   70   71    0    0    0
 18| 0 5    78   86  101  107  112    0    0   0 13 20  4 16  6 22
 19| 1 4   178  183  189  190    0    0    0
 20| 1 5    79   81   82   83   85    0    0
 21| 1 3   170  171  172    0    0    0    0
 22| 1 4   114  116  117  120    0    0    0
 23| 1 4    44   45   46   50    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 84 for incluido, qual linha,coluna na matriz?	Se o 181 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-----------------------------------------------------------------
 1| 1 6    33   35   36   37   38   39    0
 2| 0 6    40   48   58   64   75   82    0   1 21  8 15 19  4 18
 3| 1 5   138  140  143  144  145    0    0
 4| 1 3    77   78   81    0    0    0    0
 5| 1 5   168  170  171  173  176    0    0
 6| 1 7   113  115  116  117  120  121  122
 7| 1 4   180  181  184  185    0    0    0
 8| 1 6    50   53   54   55   56   57    0
 9| 1 4   148  152  154  156    0    0    0
*10| 0 2    94  137    0    0    0    0    0   2 17 11
 11| 0 6   146  157  167  179  186  193    0   3  9 20  5  7 14 22
 12| 1 5    95   96   98   99  100    0    0
 13| 1 5   127  130  132  135  136    0    0
 14| 1 4   187  188  190  191    0    0    0
 15| 1 4    59   61   62   63    0    0    0
 16| 1 5   103  105  106  108  109    0    0
 17| 0 3   101  112  126    0    0    0    0  12 16  6 13
 18| 1 4    83   85   87   92    0    0    0
 19| 1 6    65   66   69   71   72   74    0
 20| 1 4   159  160  165  166    0    0    0
 21| 1 3    41   43   44    0    0    0    0
 22| 1 3   194  195  197    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 102 for incluido, qual linha,coluna na matriz?	Se o 192 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76470 SARAH FERNANDES CHELIGA
24CC1305 - 29 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	5	3	6	7	8	10	0	0								
2	0	4	11	20	32	44	0	0	0	1	20	15	7	12			
3	1	4	128	131	132	134	0	0	0								
4	1	5	149	150	151	153	160	0	0								
5	1	6	65	73	76	79	80	83	0								
6	1	5	182	183	188	189	191	0	0								
7	1	3	34	35	37	0	0	0	0								
8	1	4	106	109	111	114	0	0	0								
9	1	6	136	137	138	140	143	145	0								
*10	0	2	64	127	0	0	0	0	0	2	17	11					
11	0	6	135	146	162	171	181	192	0	3	9	4	14	21	6	18	
12	1	7	46	49	50	55	56	57	63								
13	1	4	88	89	91	94	0	0	0								
14	1	5	164	166	167	168	169	0	0								
15	1	6	21	22	23	28	29	30	0								
16	1	7	117	118	120	122	123	125	126								
17	0	4	84	95	105	116	0	0	0	5	13	19	8	16			
18	1	5	193	195	197	199	200	0	0								
19	1	5	96	98	100	102	103	0	0								
20	1	4	14	15	17	19	0	0	0								
21	1	3	174	175	177	0	0	0	0								

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	5	3	6	7	8	10	0	0								
2	0	4	11	20	32	44	0	0	0	1	20	15	7	12			
3	1	4	128	131	132	134	0	0	0								
4	1	5	149	150	151	153	160	0	0								
5	1	6	65	73	76	79	80	83	0								
6	1	6	182	183(187)	188	189	191	0									
7	1	3	34	35	37	0	0	0	0								
8	1	4	106	109	111	114	0	0	0								
9	1	6	136	137	138	140	143	145	0								
*10	0	2	64	127	0	0	0	0	0	2	17	11					
11	0	6	135	146	162	171	181	192	0	3	9	4	14	21	6	18	
12	1	7	46	49	50	55	56	57	63								
13	1	4	88	89	91	94	0	0	0								
14	1	5	164	166	167	168	169	0	0								
15	1	6	21	22	23	28	29	30	0								
16	1	7	117	118	120	122	123	125	126								
17	0	4	84	95	105	116	0	0	0	5	13	19	8	16			
18	1	5	193	195	197	199	200	0	0								
19	1	5	96	98	100	102	103	0	0								
20	1	4	14	15	17	19	0	0	0								
21	1	3	174	175	177	0	0	0	0								

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

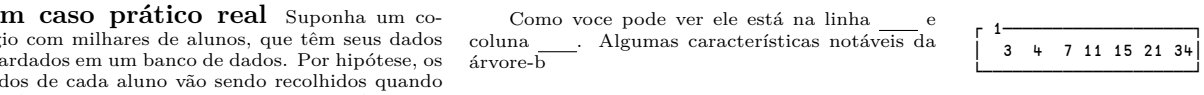
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

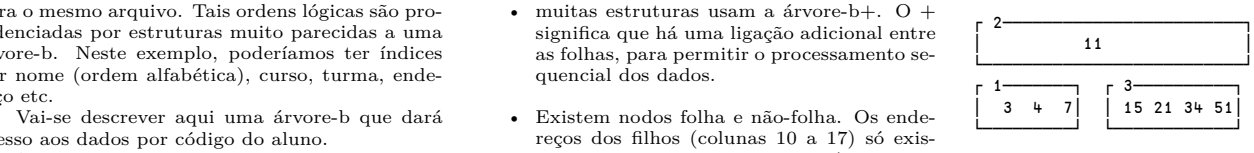
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

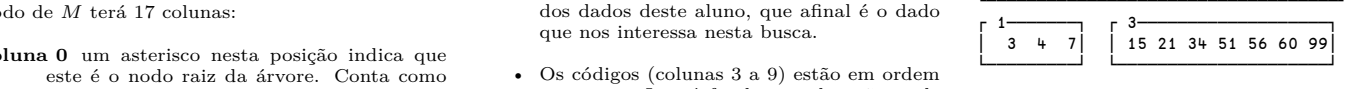


Agora, vamos inserir uma nova chave, o que vai provocar o split

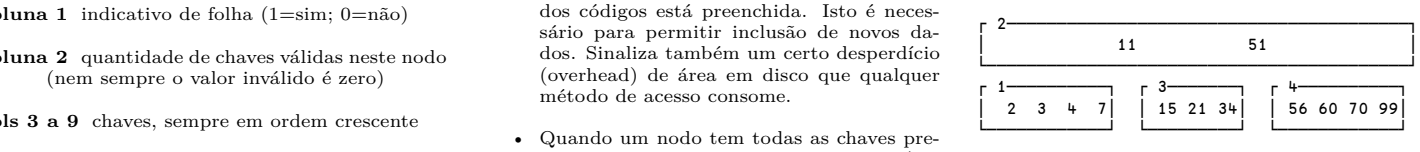
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 6    31   33   34   35   38   39    0
 2| 0 3    44   56   61    0    0    0    0  1 13 20  7
 3| 1 3    76   77   80    0    0    0    0
 4| 1 5   130  131  132  133  134    0    0
 5| 1 6    90   92   93   95  100  105    0
 6| 1 5   150  151  153  154  158    0    0
 7| 1 3    62   63   64    0    0    0    0
 8| 1 7   178  179  181  185  186  188  189
 9| 1 5   107  108  109  110  112    0    0
*10| 0 3    66  106  147    0    0    0    0  2 21 11 18
 11| 0 4   114  125  129  135    0    0    0  9 14 22  4 17
 12| 1 5    82   83   84   85   87    0    0
 13| 1 5    47   49   52   53   54    0    0
 14| 1 4   115  116  120  124    0    0    0
 15| 1 7   163  165  166  167  168  172  173
 16| 1 6   193  194  196  197  198  200    0
 17| 1 6   136  140  142  144  145  146    0
 18| 0 3   161  174  191    0    0    0    0  6 15  8 16
 19| 1 5    68   70   72   73   74    0    0
 20| 1 3    58   59   60    0    0    0    0
 21| 0 3    75   81   89    0    0    0    0 19  3 12  5
 22| 1 3   126  127  128    0    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 111 for incluido, qual linha,coluna na matriz?	Se o 43 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 5    33   34   35   36   38    0    0
 2| 0 3    40   54   63    0    0    0    0  1 13  6 17
 3| 1 3    90   92   93    0    0    0    0
 4| 1 6   151  152  153  154  155  159    0
 5| 1 4   179  180  181  185    0    0    0
 6| 1 5    56   58   59   60   61    0    0
 7| 1 4   114  115  116  117    0    0    0
 8| 1 4   164  167  168  169    0    0    0
 9| 1 7    74   75   78   80   82   83   84
*10| 0 3    73  112  160    0    0    0    0  2 23 11 19
 11| 0 4   118  128  138  149    0    0    0  7 20 12 16  4
 12| 1 5   129  130  132  134  135    0    0
 13| 1 7    43   44   47   49   50   52   53
 14| 1 4   190  192  193  194    0    0    0
 15| 1 4    95   96  102  103    0    0    0
 16| 1 5   139  140  142  143  148    0    0
 17| 1 4    67   69   71   72    0    0    0
 18| 1 4   172  173  174  176    0    0    0
 19| 0 4   170  177  189  195    0    0    0  8 18  5 14 21
 20| 1 5   120  122  123  126  127    0    0
 21| 1 4   196  197  198  199    0    0    0
 22| 1 3   108  109  110    0    0    0    0
 23| 0 3    86   94  105    0    0    0    0  9  3 15 22
```

Responda agora:

Qual a altura da árvore B ?	Se o 191 for incluido, qual linha,coluna na matriz?	Se o 66 for incluído, qual linha,coluna na matriz?

CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 05/08/2024 - 09:47:38.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Evolução de árvores: Árvore-B (e B+) VIVXq92a V: 1.02
76487 WILLIAN IELEM PAVOSKI
24CC1305 - 30 entregar ate 22/ago _____ / / _____

Árvore B

O conceito de árvore B foi desenvolvido por Rudolf Bayer e Edward McCreight em 1972.

Eles publicaram seu trabalho no artigo *"Organization and Maintenance of Large Ordered Indices"*, que descreve as propriedades e operações básicas das árvores B.

O trabalho de Bayer e McCreight teve um impacto significativo na área de ciência da computação, e as árvores B são agora amplamente utilizadas em bancos de dados e outros sistemas de software.

As árvores B são um tipo de árvore de busca balanceada que pode armazenar um grande número de chaves e valores de maneira eficiente. Elas são mais eficientes do que as árvores de busca binárias tradicionais, pois podem ter mais filhos por nó. As árvores B são usadas em uma variedade de aplicações, incluindo bancos de dados, sistemas de arquivos e sistemas de roteamento.

Eles não explicaram o significado do nome, só nos resta especular que esse B seja de Balanceada. Se você se lembrar das árvores binárias de pesquisa, deve recordar o problema que existe quando as chaves vem (quasi) ordenadas na entrada: a árvore gerada é desbalanceada e pode chegar a ter um desempenho terrível.

Outra qualidade da árvore-B é que ela tem grau k (k filhos por nodo). Em geral este valor é escolhido de modo a otimizar a blocagem de dados no arquivo de memória secundária. De qualquer modo este k modifica o desempenho, que lembrando é proporcional a $\log_k n$, onde n é a quantidade de chaves.

Como toda árvore, ela tem uma raiz que tem de 0 a $2k-1$ chaves e consequentemente tem de 0 a $2k$ filhos. Cada um dos filhos, tem de $k-1$ até $2k-1$ chaves de 0 até $2k$ filhos. Note que a raiz é o único nodo que sofre processamento especial, necessário para quando o arquivo tem uma única chave (por exemplo).

Um caso prático real Suponha um colégio com milhares de alunos, que têm seus dados guardados em um banco de dados. Por hipótese, os dados de cada aluno vão sendo recolhidos quando eles fazem a matrícula, pelo que a ordem física dos dados no arquivo é a data da matrícula. Parece óbvio assinalar que um determinado conjunto de dados (um arquivo) só pode ter uma única ordem física em um determinado instante. Mas nada impede que existam muitas ordens lógicas diferentes para o mesmo arquivo. Tais ordens lógicas são providenciadas por estruturas muito parecidas a uma árvore-b. Neste exemplo, poderíamos ter índices por nome (ordem alfabética), curso, turma, endereço etc.

Vai-se descrever aqui uma árvore-b que dará acesso aos dados por código do aluno.

Suponha uma árvore-b de nome M que providenciará este acesso. Vamos descrevê-la com $k=4$, que é muito pouco para aplicações reais, mas fica razoável para trabalhar com este exemplo. De acordo com a definição, os nodos de M têm entre 3 e 7 códigos de aluno. Para esta dimensão, o nodo de M terá 17 colunas:

coluna 0 um asterisco nesta posição indica que este é o nodo raiz da árvore. Conta como coluna 0 porque na verdade esta informação não está fisicamente aqui, está em outro lugar.

coluna 1 indicativo de folha (1=sim; 0=não)

coluna 2 quantidade de chaves válidas neste nodo (nem sempre o valor inválido é zero)

cols 3 a 9 chaves, sempre em ordem crescente

cols 10 a 17 endereços dos filhos associados a cada chave. Só estão preenchidas para nodos não folha.

Veja um exemplo:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 5  182  183  188  189  191    0    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Antes de continuar no exercício, você deve tentar entender o funcionamento da árvore-b.

- Desenhe a árvore
- identifique a necessidade da coluna 1. Idem para a coluna 2
- Qual a altura da árvore ?
- Quando e como a altura da árvore cresce ?

Agoram suponha que o elemento 187 é incluído. Qual a linha;coluna da matriz acima que é modificada, ou seja onde o elemento recém incluido entra ? Localize-o e veja:

```
    1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
  +-------------------------------------------------------------------
 1| 1 5    3    6    7    8   10    0    0
 2| 0 4   11   20   32   44    0    0    0  1 20 15  7 12
 3| 1 4  128  131  132  134    0    0    0
 4| 1 5  149  150  151  153  160    0    0
 5| 1 6   65   73   76   79   80   83    0
 6| 1 6  182  183(187)188  189  191    0
 7| 1 3   34   35   37    0    0    0    0
 8| 1 4  106  109  111  114    0    0    0
 9| 1 6  136  137  138  140  143  145    0
*10| 0 2   64  127    0    0    0    0    0  2 17 11
 11| 0 6  135  146  162  171  181  192    0  3  9  4 14 21  6 18
 12| 1 7   46   49   50   55   56   57   63
 13| 1 4   88   89   91   94    0    0    0
 14| 1 5  164  166  167  168  169    0    0
 15| 1 6   21   22   23   28   29   30    0
 16| 1 7  117  118  120  122  123  125  126
 17| 0 4   84   95  105  116    0    0    0  5 13 19  8 16
 18| 1 5  193  195  197  199  200    0    0
 19| 1 5   96   98  100  102  103    0    0
 20| 1 4   14   15   17   19    0    0    0
 21| 1 3  174  175  177    0    0    0    0
```

Como voce pode ver ele está na linha ____ e coluna ____. Algumas características notáveis da árvore-b

- Figurinha carimbada no mundo do software: usam B-trees: SYBASE e do Harbour (antigo Clipper). Também são usados no Oracle, MySQL, PostgreSQL, Microsoft SQL e DB2, da IBM.
- muitas estruturas usam a árvore-b+. O + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados.
- Existem nodos folha e não-folha. Os endereços dos filhos (colunas 10 a 17) só existem para os nodos que os têm (nodos não folhas).
- Os códigos de aluno (colunas 3 a 9 não mostram, para diminuir o ruído), mas ao lado de cada código está o endereço físico real dos dados deste aluno, que afinal é o dado que nos interessa nesta busca.
- Os códigos (colunas 3 a 9) estão em ordem crescente. Isto é fundamental e não pode ser modificado.
- Perceba como, em média, apenas a metade dos códigos está preenchida. Isto é necessário para permitir inclusão de novos dados. Sinaliza também um certo desperdício (overhead) de área em disco que qualquer método de acesso consome.
- Quando um nodo tem todas as chaves preenchidas e chega uma nova, que deveria (deverá) entrar nele, o nodo sofre um processo chamado split e ele é transformado em 2 nodos (cada um com metade dos espaços vazios). Note que o número máximo de chaves em qualquer nodo é sempre ímpar ($2n-1$) o que implica em que o elemento do meio sobe para o *pai* do nodo que está sendo quebrado. A metade anterior vai para um nodo *splitado* e a metade posterior vai para o outro nodo *splitado*.
- Quando as chaves vão sendo excluídas, e um nodo passa a ter menos do que a metade das chaves (o que é proibido) ele é juntado com um nodo irmão e ambos passam a ser um único nodo. Este processo recebe o nome de merge. Não é tão simples assim, há diversos casos particulares, mas a ideia é essa.

Mais um exemplo. Suponha a inclusão do número 26. Em que linha;coluna ele entra ?

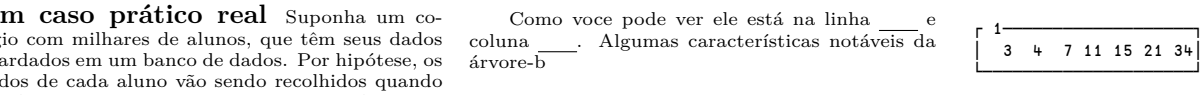
Algoritmo de Inclusão em árvore B

```
1: função INCLUA-BTREE (K:inteiro)
2: X ← raiz da árvore B
3: enquanto nodo X não é folha
4:     X ← filho correto onde K deve ser incluído
5: fim{enquanto}
6: se nodo X está cheio
7:     faça split do nodo X
8:     INCLUA-BTREE (K)
9:     abandone
10: senão
11:     inclua K no nodo X mantendo as chaves ordenadas
12: fim{se}
```

Algoritmos Estão disseminados pela comunidade. Deixo de apresentá-los aqui, pois são bastante específicos, complexos até, e mais importante do que implementá-los é entender-lhes o funcionamento. Até porque se você tiver que usar uma árvore b que não seja para aprender como ela funciona, recomendo fortemente que você use pacotes prontos, já testados e confiáveis.

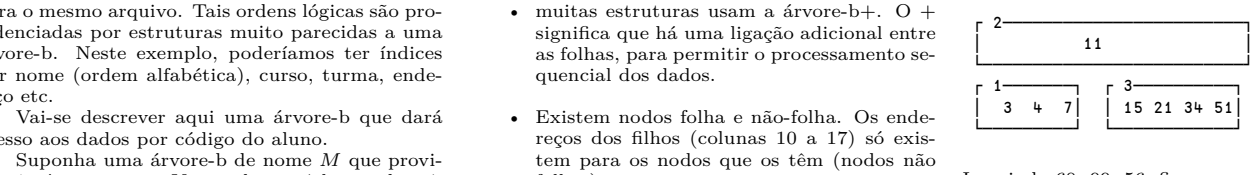
Um exemplo, passo a passo Seja uma árvore-b com $k=4$. Pela definição os nodos terão entre 3 e 7 chaves.

```
criabtree
insira 4, 7, 21, 11, 34, 15, 3
```

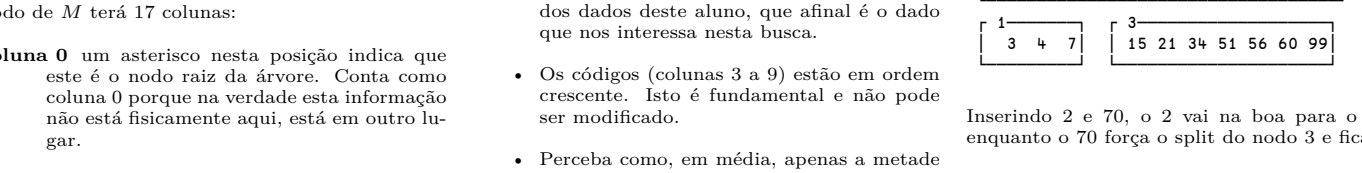


Agora, vamos inserir uma nova chave, o que vai provocar o split

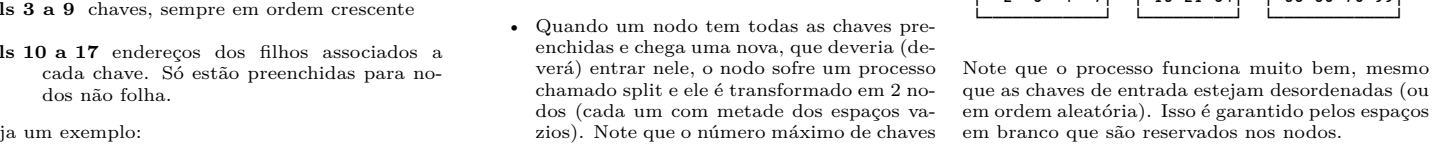
```
insira 51
```



Inserindo 60, 99, 56, fica



Inserindo 2 e 70, o 2 vai na boa para o nodo 1 enquanto o 70 força o split do nodo 3 e fica



Note que o processo funciona muito bem, mesmo que as chaves de entrada estejam desordenadas (ou em ordem aleatória). Isso é garantido pelos espaços em branco que são reservados nos nodos.

Um caso quasi-real: SYBASE

O SYBASE usa árvores B para indexar as tabelas. Existem 2 tipos de árvores B: as granuladas e as não granuladas (clustered). As granuladas são árvores esparsas (nem todo registro de dados tem entrada nos índices) nas quais os registros (na tabela original) são mantidos em ordem dos valores de índice e apenas o primeiro registro em cada página de dados tem uma entrada de índice. Árvores não granuladas são densas e cada um dos registros da tabela tem 1 entrada de índice.

Uma tabela SYBASE com índice granulado exige que seus elementos sejam mantidos em ordem dentro da tabela. Assim, inclusões podem se dar em qualquer página, a depender do conteúdo do campo de índice. Tabelas SYBASE que tem apenas índices não granulados fazem as inclusões de novos elementos sempre ao final das tabelas (na última página).

Tabelas SYBASE podem ter apenas 1 índice granulado, mas podem ter até 250 índices não granulados.

Na criação de um índice granulado, os dados da tabela devem estar colocados nas páginas em ordem.

Cada entrada de índice em sybase pode ser composta por até 16 campos e ter até 256 bytes. Estes campos não precisam estar contiguos. Só não podem ser partes de campos e resultados de cálculos.

O tamanho da página varia, mas o tamanho mais comum é 2K. Existem 5 tipos de páginas em sybase, a saber:

Páginas de Dados contém registros de dados ou registros de log. Ambos são estruturalmente diferentes, mas são construídos do mesmo jeito.

Páginas de índices contém registros de índice

Páginas de texto ou imagem contém BLOBS

Páginas de alocação contém estruturas de dados usadas para gerenciar o processo de alocacão de páginas

páginas de Estatísticas contém estatística de distribuição e uso para os índices.

Página de Dados Se os registros tem tamanho fixo, não haverá tabela de deslocamento na página e o endereço de cada registro é obtido mediante simples cálculo. Neste caso, o tamanho do registro (fixo) é guardado no campo TAMANHO MÍNIMO DE REGISTRO do header.

O header da página é um conjunto de 32 bytes contendo (entre outros): número lógico da página, anterior e próxima página lógica, identificação do objeto ao qual esta página pertence, primeira entrada de registro disponível nesta página, deslocamento do espaço livre, tamanho mínimo do registro

Depois vem a área de registros, que contém um número inteiro de slots para guardar registros. Um registro não atravessa limites de páginas

Tabela de deslocamento: Um conjunto de deslocamentos desde o início da página dando o início de cada um dos registros. Este conjunto cresce do fim da página para o início e só existe quando o tamanho do registro é variável.

Cada registro na página recebe um número de registro. É um campo de 1 byte e assim restringe-se o número de registros em uma página a 256. O número do registro usado nos índices não granulados (densos) é uma combinação do número da página com o número do registro na página.

O registro pode ocupar a página inteira (2048-32=2016). Não há tamanho mínimo para o registro, mas só pode haver 256 registros na página.

Os índices SYBASE são organizados na forma de arvores B + (o + significa que há uma ligação adicional entre as folhas, para permitir o processamento sequencial dos dados).

No exemplo aqui estudado, vão-se fazer algumas considerações/simplificações descritas a seguir.

Suponha um arquivo de pessoas contendo nome, cor preferida, bicho de estimação, alimento que gosta e idade.

Construindo um arquivo SYBASE contendo estes dados e que esteja sujeito às seguintes características:

- Registros de tamanho fixo igual a 40 bytes (9+9+9+9+4).
- Um índice granulado ou esparso (clustered) por um dos campos. No exemplo que é igual para todos, o índice esparso é por cor.
- Quatro índices não granulados (densos) pelos demais atributos.
- Páginas de dados com espaço para 4 registros e fator de preenchimento de 4:1 (usar apenas 3 espaços na carga original).
- Páginas de índices com espaço para 5 chaves (nome, cor, bicho, alimento ou idade). Lembrar que este número é impar, o que indica que no *split* vai subir a terceira chave (das 5 que existem).
- A identificação de cada registro deve ser página:número do registro na página. Por exemplo, a identificação 3:2, indica o segundo registro da terceira página.
- Nas páginas de índice (árvores-B) despreze as informações de endereço dos filhos. No exemplo, elas são irrelevantes. Faça como no exemplo.
- Os dados de entrada estão ordenados pela coluna do índice granulado.

Exemplo: Seja o conjunto de dados, no qual se fará o granulado por cor:

NOME	COR	ANIMAL	COMIDA	id.
josé	dourado	chopim	feijão	34
gabi	laranja	perú	aipim	22
sueli	lilas	gato	alho	40
antônio	musgo	vaca	batata	41
tatiana	petróleo	cavalo	arroz	37
juca	preto	águia	goiaba	36
romeu	roxo	dragão	maracujá	27
paulo	verde	urubú	rabanete	20

Com este conjunto de dados, ter-se-á a seguintes disposição nos arquivos

Páginas de Dados					
Pág1	josé	dourado	chopim	feijão	34
	gabi	laranja	perú	aipim	22
	sueli	lilas	gato	alho	40
Pág2	antônio	musgo	vaca	batata	41
	tatiana	petróleo	cavalo	arroz	37
	juca	preto	águia	goiaba	36
Pág3	romeu	roxo	dragão	maracujá	27
	paulo	verde	urubú	rabanete	20

A seguir, as páginas de índices

Páginas de índices	
Índice I1	granulado por COR
Página 1 ←raiz	dourado(1:1), musgo(2:1), roxo(3:1)
Índice I2	por NOME
Página 1	antônio(2:1), gabi(1:2)
Página 2 ←raiz	jose(1:1)
Página 3	juca(2:3), paulo(3:2), romeu(3:1), sueli(1:3), tatiana(2:2)
Índice I3	por BICHO
P1	águia(2:3), cavalo(2:2), chopim(1:1), dragão(3:1)
P2 ←raiz	gato(1:3)
P3	perú(1:2),urubú(3:2), vaca(2:1)
Índice I4	por COMIDA
P1	aipim(1:2),alho(1:3)
P2 ←raiz	arroz(2:2)
P3	batata(2:1), feijão(1:1), goiaba(2:3), maracujá(3:1), rabanete(3:2)
Índice I5	por IDADE
P1	20(3:2), 22(1:2), 27(3:1), 34(1:1), 36(2:3)
P2 ←raiz	37(2:2)
P3	40(1:3), 41(2:1)

☞ Para você fazer

Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 4    32   33   37   41    0    0    0
 2| 0 3    42   51   63    0    0    0    0  1 14  6  9
 3| 1 7   168  169  171  172  173  175  178
 4| 1 5    78   79   81   82   83    0    0
 5| 1 5   123  125  127  130  131    0    0
 6| 1 6    52   54   56   59   61   62    0
 7| 1 3   104  105  106    0    0    0    0
 8| 1 6   182  183  185  186  188  189    0
 9| 1 4    64   65   66   67    0    0    0
*10| 0 3    68  102  148    0    0    0    0  2 19 11 21
 11| 0 4   110  122  132  138    0    0    0  7 20  5 16 23
 12| 1 5   149  150  151  152  153    0    0
 13| 1 3    87   88   89    0    0    0    0
 14| 1 6    43   44   45   47   48   49    0
 15| 1 5   194  196  197  198  200    0    0
 16| 1 3   133  135  137    0    0    0    0
 17| 1 4    70   72   74   75    0    0    0
 18| 1 5    91   94   95   97  100    0    0
 19| 0 3    77   85   90    0    0    0    0 17  4 13 18
 20| 1 5   111  113  114  116  117    0    0
 21| 0 4   155  167  181  190    0    0    0 12 22  3  8 15
 22| 1 3   161  165  166    0    0    0    0
 23| 1 4   139  140  145  146    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 160 for incluido, qual linha,coluna na matriz?	Se o 98 for incluído, qual linha,coluna na matriz?

Mais uma Instância do mesmo exercício Desenhe a árvore B a seguir:

```
     1 2    3    4    5    6    7    8    9 10 11 12 13 14 15 16 17
   +-------------------------------------------------------------------
 1| 1 7    31   32   33   34   35   36   37
 2| 0 6    38   49   66   75   90  105    0  1  7 12  4 14  8 19
 3| 1 4   135  137  138  139    0    0    0
 4| 1 5    68   70   71   72   74    0    0
 5| 1 7   114  115  118  120  122  124  127
 6| 1 3   166  169  170    0    0    0    0
 7| 1 6    39   43   44   45   46   48    0
 8| 1 4    91   93   97   98    0    0    0
 9| 1 6   183  184  185  186  190  192    0
*10| 0 2   113  153    0    0    0    0    0  2 11 18
 11| 0 4   128  134  140  145    0    0    0  5 15  3 20 13
 12| 1 6    52   53   54   56   59   63    0
 13| 1 4   147  148  151  152    0    0    0
 14| 1 7    77   78   79   80   81   82   83
 15| 1 5   129  130  131  132  133    0    0
 16| 1 4   195  196  198  199    0    0    0
 17| 1 5   154  156  158  160  161    0    0
 18| 0 4   162  172  182  194    0    0    0 17  6 21  9 16
 19| 1 4   106  107  108  110    0    0    0
 20| 1 3   141  142  143    0    0    0    0
 21| 1 4   176  177  179  180    0    0    0
```

Responda agora:

Qual a altura da árvore B ?	Se o 100 for incluido, qual linha,coluna na matriz?	Se o 157 for incluído, qual linha,coluna na matriz?