

Exercício de Árvores B com grau t=3

Em 1972, Bayer e MacCreight introduziram uma estrutura de dados bastante adequada para resolver o problema de construir índices em memórias secundárias. Denominaram-na B-árvore, mas abstiveram-se de explicar este nome. A B-árvore é uma árvore k-ária, isto é, cada nodo pode ter k filhos, sendo que em geral este valor de k é escolhido de tal maneira a otimizar a blocagem física do arquivo de índices.

O nodo original é chamado "raiz" e é o único que não tem pai, mas pode ter de 0 até $2n - 1$ chaves e $2n$ filhos. Cada um dos filhos, por sua vez, pode ter de $n - 1$ a $2n - 1$ chaves e de 0 a $2n$ filhos (que serão netos do nodo raiz) e assim sucessivamente. Este valor de n, é denominado grau da árvore, e as árvores B têm este valor na casa das dezenas ou centenas. Cria-se assim uma estrutura piramidal, com vaga semelhança com as árvores: daí o nome.

Seja o problema de construir uma matriz M representando uma árvore B (BTREE) sujeita às seguintes condições:

1. Os nodos comportam 5 chaves e 6 filhos cada
2. Os nodos são linhas de uma matriz alocada seqüencialmente
3. As colunas da matriz são:

coluna 1 1 se esta linha é folha e 0 senão

coluna 2 contador de chaves válidas neste nodo

cols 3-4-5-6-7 chaves (ou zero)

cols 8-9-10-11-12-13 se o nodo não é folha, o cursor dos filhos

4. A árvore B mantém os dados em ordem crescente
5. A raiz é representada por um "*"

Nesta estrutura, use os seguintes algoritmos

```
#include<stdio.h>
int barv[100][14];
int dr;
int dprox;
int dt;
int insbarv (int chave) {
    int r,s;
    r = dr;
    if (barv[dr][2]>=((2*dt)-1)) {
        s = dprox;
        dprox = dprox+1;
        dr=s;
        barv[s][1]=0;
        barv[s][2]=0;
        barv[s][2+(2*dt)]=r;
        quebrabarv(s,1,r);
        insncbarv(s,chave);
    }
    else {
        insncbarv(r,chave);
    }
}

int quebrabarv(int pai, int i, int filho){
    int z,j;
    z = dprox;
    dprox = dprox + 1;
    barv[z][1]=barv[filho][1]; //folha=1 ou nao=0
    barv[z][2]=dt-1; // quantidade de chaves no coneeo (t-1)
    for (j =1; j<dt;j++){ //repassando as chaves para o filho
        barv[z][2+j] = barv[filho][2+dt+j];
    }
    if (barv[filho][1]==0){ //se tem pointers, tem que move-los
        for (j=1; j<dt;j++){
            barv[z][(((2*dt)+2)-1)+j]=barv[filho][(((2*dt)+2)-1)+dt+j];
        }
    }
    barv[filho][2]=dt-1; // quantidade de chaves do filho
    for (j=barv[pai][2]+1;j!=1;j--){
        barv[pai][2+(2*dt))+j]=barv[pai][(((2*dt)+2)-1)+j];
    }
    barv[pai][2+(2*dt))+1=j=z;
    for (j=barv[pai][2];j>i-1;j--){
        barv[pai][3+j]=barv[pai][2+j];
    }
    barv[pai][2+i]=barv[filho][2+dt];
    barv[pai][2]=barv[pai][2]+1;
}

int insncbarv(int nodo, int chave){ // no apl=funcao nc
```

```
int i,xvelho;
i = barv[nodo][2];
if (barv[nodo][1]!=0){
    while ((i >= 1) && (chave < barv[nodo][2+i])) {
        barv[nodo][i+3]=barv[nodo][i+2];
        i--;
    }
    barv[nodo][i+3]=chave;
    barv[nodo][2] = barv[nodo][2]+1;
}
else {
    while ((i >= 1) && (chave < barv[nodo][2+i])) {
        i--;
    }
    i++;
    xvelho=nodo;
    nodo=barv[nodo][(((2*dt)+2)-1)+1];
    if (barv[nodo][2]==((2*dt)-1)) {
        quebrabarv (xvelho,1,nodo);
        if (chave > barv[xvelho][2+i]) {
            i++;
        }
    }
    insncbarv (barv[xvelho][(((2*dt)+2)-1)+i], chave);
}
}

int printabarv() {
    int k;
    for (k=1;k<dprox;k++){
        printf ("%2i %2i %3i %3i %3i %3i %3i %3i %3i\n",barv[k][1],barv[k][2],barv[k][3],barv[k][4],barv[k][5],barv[k][6],barv[k][7],barv[k][8],barv[k][9],barv[k][10],barv[k][11],barv[k][12],barv[k][13],barv[k][14]);
    }
    printf("a raiz e = %i\n",dr);
}

main() {
    int i,j,k;
    for (i=0;i<100;i++){
        for(j=0;j<14;j++){
            if (j==1) {
                barv[i][j]=1;
            }
            else {
                barv[i][j]=0;
            }
        }
    }
    dt=3; // alterar este numero se quiser arvores diferentes
    dprox=2;
    dr=1;
    insbarv(20);
    insbarv(42);
    ...
    insbarv(4);
    printabarv();
}
```

Seja um exemplo

Criar uma matriz vazia e inserir os números:

16	65	89	79	42	35	52	37	2	71	46	29	14	22	40	80	92
44	57	49	32	73	11	47	38	10	77	68	41	12	78	63	17	
96	5	7	50	75	28											

A matriz resultado será:

		1	2		3		4		5		6		7		8		9		10		11		12		13
	+																								
1		1	5		2		5		6		7		10												
2		0	4		11		16		29		37		0		1	12		5	13		4				
3		1	2		68		71		0		0		0												
4		1	4		38		40		41		42		0												
5		1	3		17		22		28		0		0												
6		1	2		46		47		0		0		0												
7		1	3		89		92		96		0		0												
* 8		0	1		44		0		0		0		0		2	9									
9		0	4		49		65		73		80		0		6	11		3	10		7				
10		1	4		75		77		78		79		0												
11		1	4		50		52		57		63		0												
12		1	2		12		14		0		0		0												
13		1	2		32		35		0		0		0												

As perguntas e as respostas:

1. Qual a primeira chave do nodo raiz ? R1: _____.
2. Quantas são as linhas de M que tem 3 chaves válidas ? R2: _____.
3. Quanto é a soma vertical da coluna 8 de M ? R3: _____.
4. O valor da última chave válida na linha 3 de M é: R4: _____.

🔖 Para você fazer

Os 40 numeros a incluir são

89	18	65	85	80	15	51	10	52	63	78	17	46	86
44	50	8	27	49	62	88	34	93	11	4	43	31	84

Para ajudar, segue um esqueleto de árvore B

	F?	C?	K_1		K_2	K_3	K_4	K_5	C_1	C_2	C_3	C_4
	1	2	3	4	5	6	7	8	9	10	11	
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												

Construa a árvore e responda as questões:

Qual a primeira chave do nodo raiz ?	Quantas são as linhas de M que tem 3 chaves válidas ?	Quanto é a soma vertical da coluna 8 de M ?	O valor da última chave válida na linha 6 de M é

Bibliografia para esta folha: ALGORITMOS, Cormen et alli.

