

👉 Para você fazer

Resolva a seguinte instância
 Crie as 3 árvores com os elementos a seguir (obs: antes de começar, conte todos os elementos: eles devem totalizar 60 números nas 3 árvores).

ABP

A estrutura de árvore binária de pesquisa (conhecida como ABP) é muito importante na ciência da computação. Baseado nela existe um dos mais eficientes e simples algoritmos de ordenação: cria-se a abp com os elementos a ordenar e depois percorre-se a árvore em ordem: estão ordenados todos os elementos. Tem uma eficiência de $O(n \times \log_2 n)$.

Lembrando, uma ABP é binária (no máximo 2 filhos) e na sua construção usa-se a regra, que é válida para todos os seus nodos: tomando o nodo x como raiz, todos os nodos ligados ao filho esquerdo deverão ter valor MENOR que x , e todos os nodos ligados ao filho direito deverão ter valor MAIOR que x .

Esta folha vai tratar os algoritmos de criação, inclusão e exclusão de valores em *abps*. Em nenhum momento haverá duplicidade de chaves em qualquer árvore, isto não precisa ser testado.

Finalmente, depois pede-se um caminhamento sobre a árvore. A seguir os principais caminhamentos. Lembrar que os 3 primeiros devem ser aplicados recursivamente a toda a árvore.

pré-ordem raiz, filho-esquerdo, filho-direito.

em-ordem filho-esquerdo, raiz, filho-direito.

pós-ordem filho-esquerdo, filho-direito, raiz.

largura nível 0 (raiz), depois todos os do nível 1, 2,

Inclusões e Exclusões em ABPs

Suponha 3 ABPs independentes, com cerca de 20 números cada uma. Haverá uma seqüência de 20 operações de exclusão em uma das árvores seguida da inclusão do mesmo elemento em outra das duas árvores.

Ao final, continuará havendo as 3 árvores, com 20 elementos tendo trocado de árvore. Você vai receber a seqüência de criação das 3 árvores e depois deve efetuar as 20 operações. Depois que as 3 árvores finais estiverem prontas, você deve produzir a visitação horizontal das 3 árvores. Finalmente, deve responder a perguntas sobre essa visitação.

Lembre-se que a ordem dos elementos na entrada precisa ser rigorosamente obedecida e não pode ser alterada, sob pena do formato da árvore (e consequentemente a visitação) ser alterada.

Ao incluir, recorde-se que o local de inclusão é único e está sempre perfeitamente determinado, desde que se obedeça sempre a regra geral que é: "dado um elemento na raiz, (todos) os filhos esquerdos terão que ser MENORES do que a raiz e (todos) os filhos direitos terão que ser MAIORES do que a raiz.

Ao excluir, considere os 3 casos de exclusão, a depender do número de filhos do nodo em questão:

sem filhos A exclusão é trivial, basta eliminar o nodo e corrigir o apontamento do pai para ele.

1 filho O filho do nodo a excluir substitui o nodo a excluir.

2 filhos Este é o caso mais complicado. Há algumas tarefas

- Localizar na árvore o nodo sucessor do nodo a excluir
- Excluir o sucessor
- Trocar o nodo a excluir pelo seu sucessor

Algumas observações: 1. O sucessor nunca terá os 2 filhos, daí não existe a possibilidade do algoritmo entrar em loop. 2. Fazendo a troca do sucessor pelo nodo a excluir, por mais distante que o sucessor esteja do sucedido, a árvore sempre continua sendo uma árvore ABP.

Exemplo Sejam as seguintes 3 árvores:

```
35 78 16 55 33 8 29 91 56 80 15 86 87 44 43
11 47 48 67 89 39 45 28 68 41 69 3 85 46
31 7 90 1 83 26 93 32 73 66 9 60 12 2 70 14
```

A seguir, devem-se executar as seguintes operações: (a ordem das operações não é relevante).

```
retire 09 de 3 e insira-o em 1
retire 29 de 1 e insira-o em 2
retire 60 de 3 e insira-o em 2
retire 32 de 3 e insira-o em 1
retire 73 de 3 e insira-o em 2
retire 46 de 2 e insira-o em 1
retire 67 de 2 e insira-o em 1
retire 15 de 1 e insira-o em 3
retire 02 de 3 e insira-o em 1
retire 45 de 2 e insira-o em 3
retire 93 de 3 e insira-o em 1
retire 48 de 2 e insira-o em 3
```

Após a execução destas operações, o caminhamento em largura das 3 árvores resultantes ficou assim:

```
35 16 78 8 33 55 91 2 9 32 44 56 80 93 43 46 67 86 87
11 3 47 39 68 28 41 60 89 29 69 85 73
31 7 90 1 26 83 12 66 14 45 70 15 48
```

Árvore 1:

43 94 37 18 4 86 91 40 71 24 17 5 79 64 84 58 42 33 83

Árvore 2:

27 53 6 69 70 76 90 47 41 60 61 96 23 95 93 32 74 38 98 3

Árvore 3:

26 29 80 92 19 13 35 97 82 73 9 88 8 10 20 15 89 7 56 44 48

Depois aplique as seguintes operações, em número de 20:

```
retire 97 de 3 e insira-o em 1
retire 48 de 3 e insira-o em 2
retire 03 de 2 e insira-o em 3
retire 91 de 1 e insira-o em 2
retire 29 de 3 e insira-o em 1
retire 15 de 3 e insira-o em 2
retire 17 de 1 e insira-o em 2
retire 06 de 2 e insira-o em 3
retire 69 de 2 e insira-o em 1
retire 24 de 1 e insira-o em 2
retire 53 de 2 e insira-o em 1
retire 32 de 2 e insira-o em 1
retire 86 de 1 e insira-o em 2
retire 37 de 1 e insira-o em 3
retire 10 de 3 e insira-o em 2
retire 61 de 2 e insira-o em 1
retire 98 de 2 e insira-o em 3
retire 96 de 2 e insira-o em 3
retire 90 de 2 e insira-o em 1
retire 09 de 3 e insira-o em 2
```

E agora construa o VETOR DE CAMINHAMENTO EM LARGURA das 3 árvores e depois responda:

Arv 1	Arv 1	Arv 2	Arv 3	Arv 3
V[14]	V[17]	V[14]	V[19]	V[12]

Para esta folha, a origem dos índices é 1. Logo, a contagem é 1, 2, 3...n e não 0, 1, 2...n - 1.

1

||||| ||||| ||||| - 1 - /