

# AED2 2022 (1s) - EXERCÍCIO 8

---

## Instruções:

1. E/S: tanto a entrada quanto a saída de dados devem ser "secas", ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas (veja o exemplo abaixo);
2. Identificadores de variáveis: escolha nomes apropriados;
3. Documentação: inclua cabeçalho, comentários e indentação no programa;
4. Submeta o programa no sistema judge <http://judge.unifesp.br/aed2S01A2022/>;
5. O código-fonte pode ser escrito em C, C++, Java.

**Descrição:** Implemente as operações básicas de um algoritmo de pesquisa para árvores do tipo AVL. Seu programa **deve** conter os seguintes procedimentos: (1) inicialização, (2) pesquisa, (3) inserção, (4) remoção e (5) calcular altura do nó. Para isso, considere:

- Na operação de remoção, quando necessário, dê a preferência para a promoção da menor chave da subárvore à direita, ou seja, o sucessor;
- A altura de um nó  $x$  em uma AVL é a mesma já utilizada na atividade anterior, sobre ABB. Isto é, a distância entre  $x$  e o seu descendente mais afastado, ou seja, a altura de  $x$  é o número de passos no mais longo caminho que leva de  $x$  até um nó folha.

Lembre-se que, cada inserção em uma árvore AVL requer que se verifique o fator de balanceamento e, se necessário, uma rotação é realizada. As remoções também podem demandar rotações para manter a árvore balanceada.

Considere as seguintes condições:

1. A complexidade de cada procedimento implementado deve a mesma apresentada em aula;
2. O código-fonte **deve** ser escrito em C/C++ ou Java;
3. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;

Solução que violem essas condições **não** serão aceitas.

## ENTRADA:

A primeira linha da entrada consiste de uma lista de números inteiros positivos separados por espaços. Esses números devem ser inseridos na árvore. A sequência termina com um número inteiro negativo que não deve ser inserido.

A segunda linha contém outra lista de números inteiros positivos separados por espaços. Assim como na lista da primeira linha, um número inteiro negativo marca o final da lista. Cada um desses números, com exceção do número negativo, deve-se ser usado como chave de pesquisada na árvore. No caso de uma pesquisa sem sucesso, ou seja, o número não está na árvore, o número pesquisado deverá ser inserido na árvore.

Se a pesquisa for bem sucedida, ou seja, o número está na árvore, o número deverá ser removido.

A terceira linha contém um único valor inteiro positivo a ser apenas **pesquisado** na árvore. Esse número pode ou não estar na árvore.

### S A Í D A:

A primeira linha da saída contém a altura máxima da ABB a partir do seu nó raiz seguida da altura da sub-árvore da esquerda e da direita do nó raiz. Esses valores devem ser calculados considerando apenas a árvore construída com os números da primeira linha da entrada.

Na segunda linha, deve ser impresso o valor da altura do nó pesquisado (linha 03 dos dados de entrada), seguido pela altura da sub-árvore da esquerda e direita. Caso esse valor pesquisado não seja encontrado, deve ser exibido "Valor nao encontrado".

### Exemplos de entrada e saída:

- *input01*:

Entrada	Saída
6 4 3 2 1 -1	2, 2, 1
4 5 -1	0, 0, 0
6	

Tabela 1: Exemplos de entrada e saída 01.

A Figura 1 exibe a árvore AVL referente a entrada de dados constante da linha um do primeiro exemplo, onde a altura a partir do nó raiz é  $h = 2$ , com altura da esquerda  $he = 2$  e da direita  $hd = 1$ .

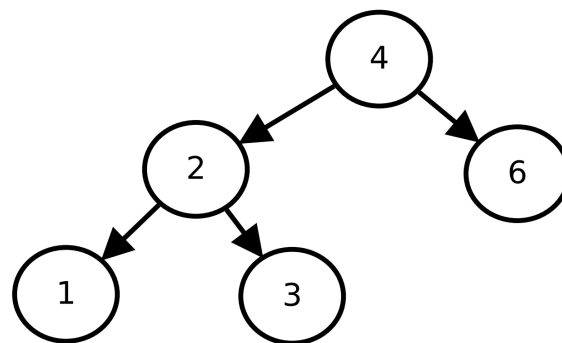


Figura 1: ABB do tipo AVL depois da inserção de todos elementos de entrada.

Na Figura 2, tem-se a árvore binária do tipo AVL, onde o nó de valor 4 deverá ser removido e substituído pelo seu sucessor, o nó de valor 6.

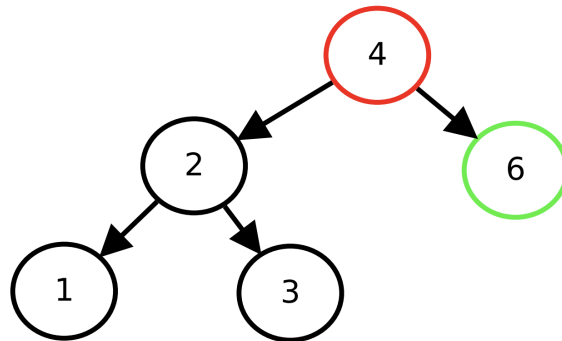


Figura 2: ABB do tipo AVL com destaques para o nó 4 (borda vermelha) a ser removido e para o nó 6 (borda verde) que é o seu do nó 4.

Na Figura 3, tem-se uma árvore desbalanceada após a última remoção (nó quatro), com destaque para o nó de valor 6 ocupando a raiz da árvore.

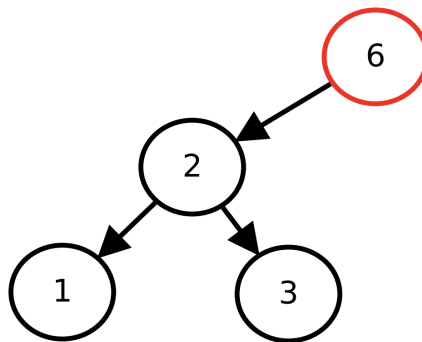


Figura 3: ABB desbalanceada após a remoção do 4.

Na Figura 4, tem-se a árvore já balanceada após rotação a direita.

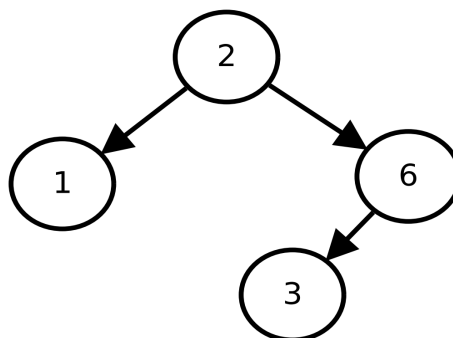


Figura 4: ABB após rotação a direita aplicada na raiz.

Na Figura 5, tem-se a árvore desbalanceada após a inserção do nó 5.

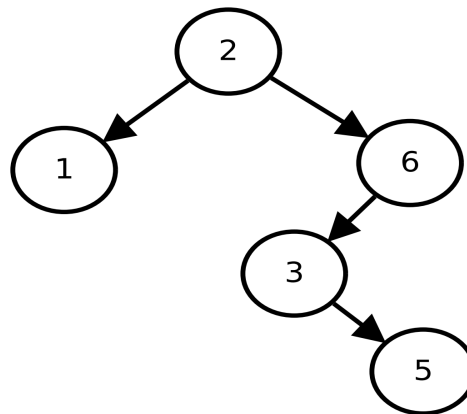


Figura 5: ABB desbalanceada após a inserção do valor 5.

Na Figura 6, tem-se a árvore final balanceada após a última inserção (5).

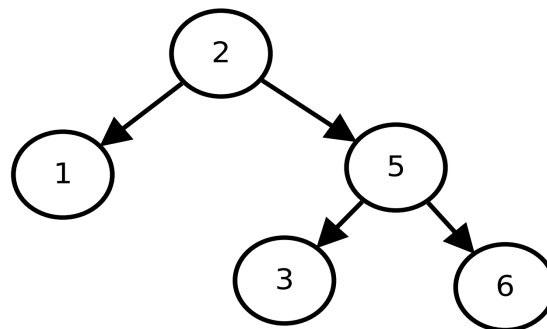


Figura 6: ABB após o balanceamento.

Ao pesquisar pelo nó de valor 6, verifica-se que sua altura é  $h = 0$ ,  $he = 0$  e  $hd = 0$ , pois se trata de um nó folha, portanto sem filhos.

- *input02*

No exemplo 2, a ABB do tipo AVL, formada pelos dados de entrada da Tabela 2, é a mesma do *input01*, conforme se observa nas Figuras 6 e 7. Entretanto, a figura 7 tem destaque para o nó de valor 5 (linha 03 da entrada), para o qual será feito uma pesquisa, verificando que sua altura é  $h = 1$ ,  $he = 1$  e  $hd = 1$ .

Entrada	Saída
6 4 3 2 1 -1	2, 2, 1
4 5 -1	1, 1, 1
5	

Tabela 2: Exemplos de entrada e saída 02.

- *input03*:

A Figura 8 exibe a árvore AVL referente aos dados *input 03*, dispostos na Tabela 3. A altura da árvore é  $h = 3$ , com à esquerda  $he = 3$  e à direita  $hd = 3$ .

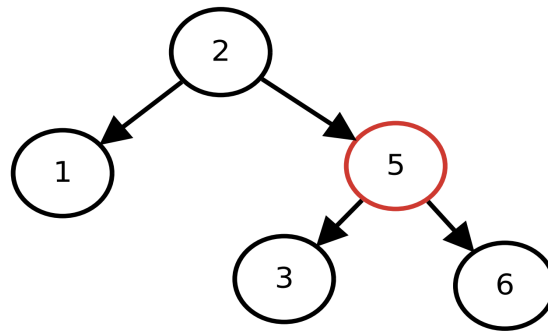


Figura 7: Busca pelo elemento 5.

Entrada	Saída
16 14 20 12 11 19 18 15 17 13 -1	3, 3, 3
14 19 15 20 -1	1, 1, 0
18	

Tabela 3: Exemplos de entrada e saída 03

A Figura 9 exibe a ABB do tipo AVL após as inserções e remoções solicitadas na linha 02 do *input 03*.

A Figura 10 exibe a ABB do tipo AVL com destaque para o o nó de valor 18, que será pesquisado, conforme informado na linha 03 do *input 03*. A altura desse nó é  $h = 1$ , com à esquerda  $he = 1$  e à direita  $hd = 0$ .

- *input04*:

Entrada	Saída
16 14 20 12 11 19 18 15 17 13 -1	3, 3, 3
14 19 15 20 -1	Valor nao encontrado
100	

Tabela 4: Exemplos de entrada e saída 03

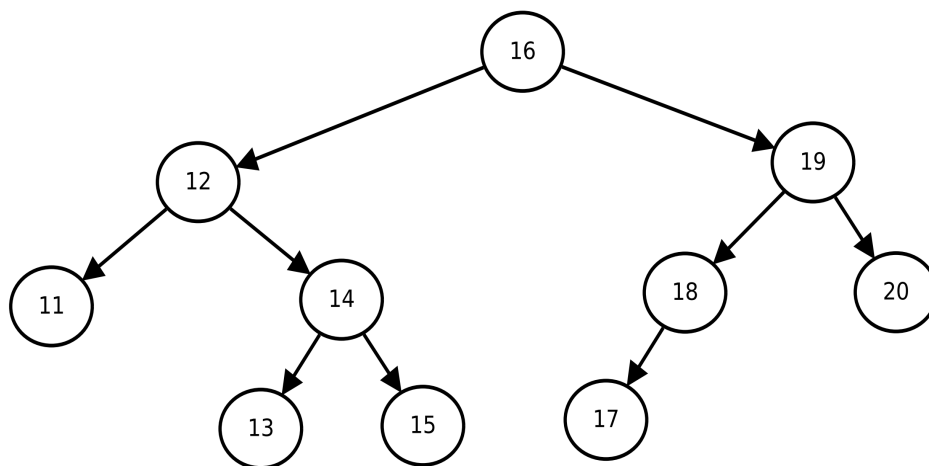


Figura 8: AVL após a inserção dos elementos constantes da segunda linha dos dados de entrada do input3.

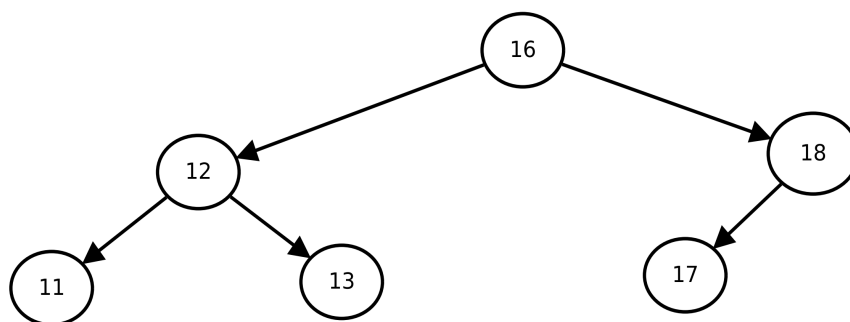


Figura 9: Árvore AVL depois da inserção e remoção dos elementos do input03.

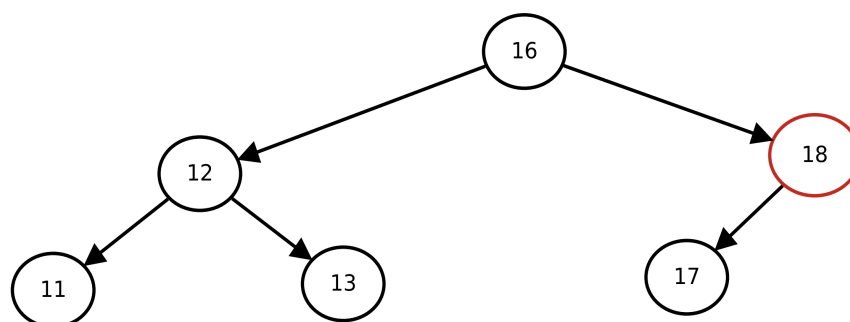


Figura 10: ABB com destaque para a busca do elemento 18.