

# AED2 2021 - TAREFA 9 - ÁRVORE AVL

Entrega: 26/05/2021 até 23:59:59

---

## 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 utilizando acesso remoto via VPN: `http://judge.sjc.unifesp.br/aed2`, ou através de conexão direta: `http://kp.unifesp.br:9001/aed2/login`.
5. O código-fonte pode ser escrito em C, C++ ou Java.

**Descrição:** Implemente as operações básicas de um algoritmo de pesquisa para Árvores AVL conforme as especificações abaixo. Escreva um programa que contenha os procedimentos a seguir: (1) inicialização, (2) pesquisa, (3) inserção e (4) remoção (5) calculo da altura do nó. Não é permitido o uso de algoritmos de ordenação para manter os dados ordenados, caso seja necessário. 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 distância entre  $x$  e o seu descendente mais afastado, ou seja, a altura de  $x$  é o número de **nós percorridos na árvore** (ou passos) no mais longo caminho, a partir do nó  $x$  até o nó folha mais distante. Lembre-se que na árvore AVL a inserção já é balanceada, ou seja, a cada inserção checa-se o fator de balanceamento e, se necessário, é feita uma rotação. A definição de altura é a mesma já utilizada na atividade anterior. Obs: As entradas e saídas são semelhantes a atividade anterior, o que muda nesta atividade é o fato das inserções e remoções demandarem possíveis rotações para manter a árvore do tipo AVL balanceada.

## ENTRADA:

A primeira linha da entrada para criação da árvore, consiste de uma sequência de números inteiros positivos separados por espaços. A sequência termina quando for digitado um número inteiro negativo. Essa sequência inicial consiste de valores a serem inseridos na árvore binária do tipo AVL.

A linha seguinte contém outra sequência de números inteiros positivos separados por espaços, com um número inteiro negativo no final, onde cada valor será utilizado como chave a ser pesquisada na árvore. No caso de uma pesquisa sem sucesso, esse número deverá ser inserido no dicionário, caso contrário, ele deverá ser removido.

A terceira linha representa um único valor inteiro positivo a ser apenas pesquisado na árvore, o qual pode ou não existir.

## SAÍDA:

A primeira linha da saída refere-se a árvore inicial gerada pela primeira linha dos dados de entrada, e representa a altura máxima da ABB do tipo AVL a partir do seu nó raiz, seguido da altura da sub-árvore da esquerda e da direita do nó raiz.

Na linha seguinte deve ser impresso o valor da altura do nó pesquisado (linha 03 dos dados de entrada), seguido pela altura 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

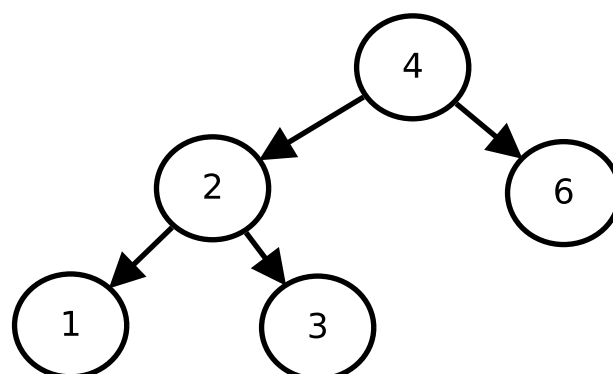


Figura 1: ABB do tipo AVL depois da inserção de todos elementos de entrada

A Figura 1 exibe a árvore AVL referente a entrada de dados constante da linha 01 do primeiro exemplo, onde a altura a partir do nó raiz é  $h = 2$ , com altura da esquerda  $he = 2$  e da direita  $hd = 1$ .

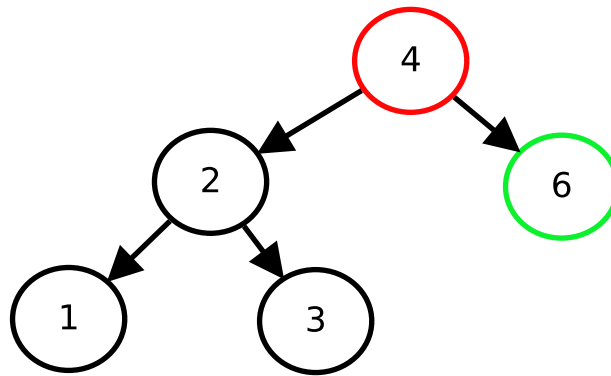


Figura 2: ABB do tipo AVL com destaques para o nó 4, com borda vermelha, a ser removido, e para o nó 6 que representa o seu sucessor

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

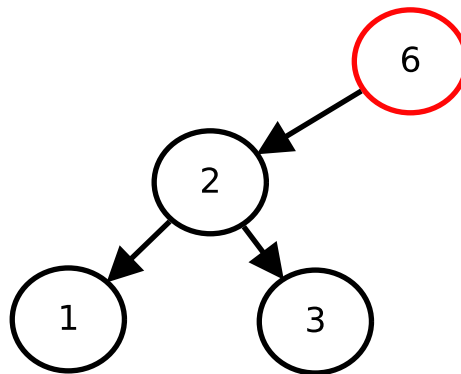


Figura 3: ABB desbalanceada após a remoção do 4

Na Figura 3 tem-se uma árvore desbalanceada após a última remoção, com destaque para o nó de valor 6.

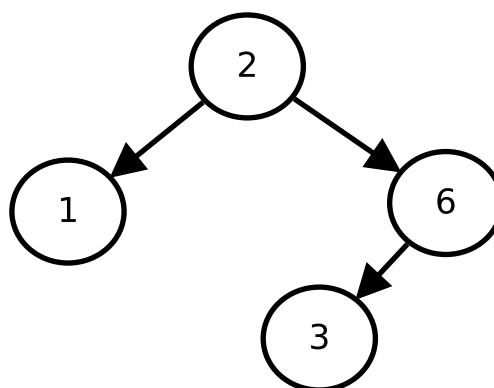


Figura 4: ABB após rotação à direita aplicada na raiz

Na Figura 4 tem-se a árvore já balanceada com rotação a direita, após a remoção do nó 4.

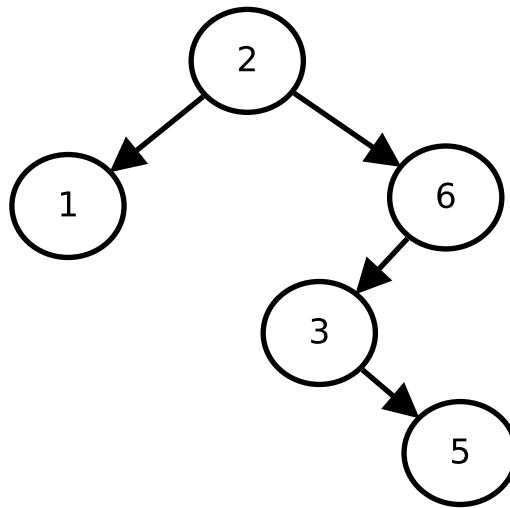


Figura 5: ABB desbalanceada após a inserção do valor 5

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

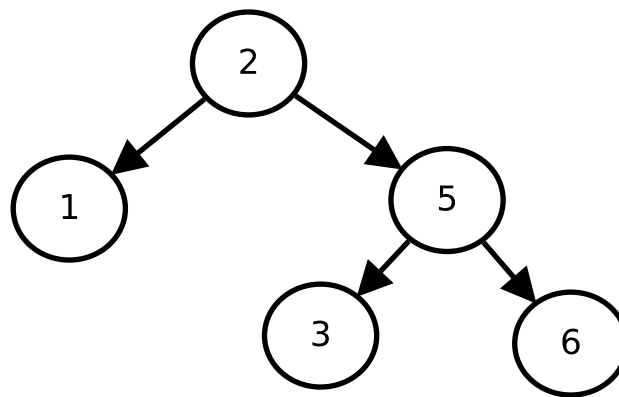


Figura 6: ABB após o balanceamento

Na Figura 6 tem-se a árvore final balanceada após a última inserção.

Ao pesquisar o nó de valor 6, encontra-se o mesmo com altura  $h = 0$ ,  $he = 0$  e  $hd = 0$ , pois se trata de um nó folha, portanto sem filhos.

- *input02*

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

No exemplo 2, a ABB do tipo AVL formada após as linhas 01 e 02 dos dados de entrada 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, encontrando  $h = 1$ ,  $he = 1$  e  $hd = 1$ .

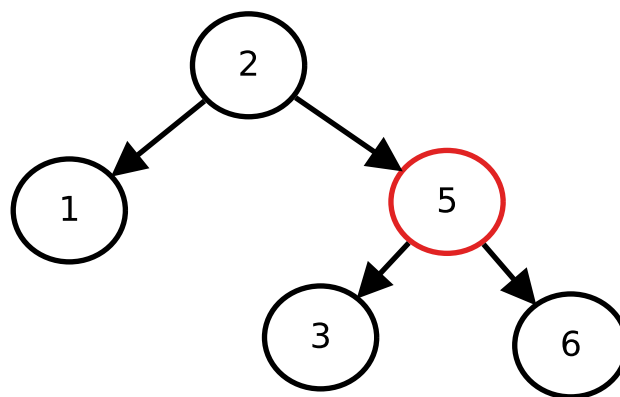


Figura 7: Busca do elemento 5

- *input03*:

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

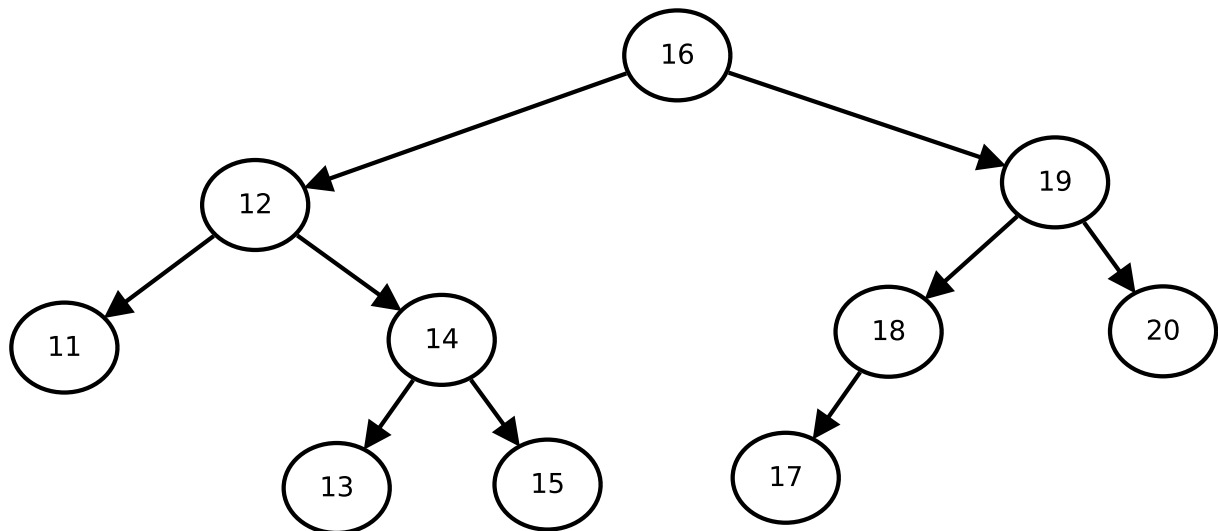


Figura 8: Árvore AVL depois da inserção dos elementos referentes a linha 01 de entrada do exemplo 03

A Figura 8 exibe a árvore AVL referente a entrada da linha 01 do *input 03*, onde sua altura  $h = 3$ , com à esquerda  $he = 3$  e à direita  $hd = 3$ .

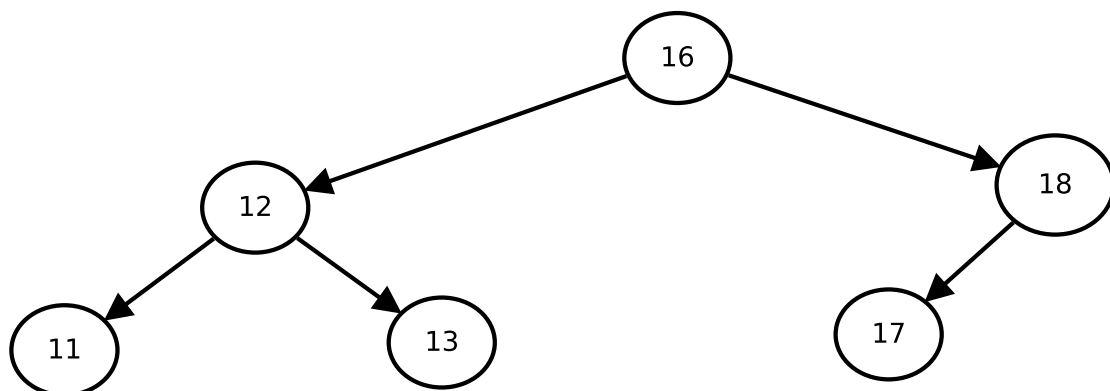


Figura 9: Árvore AVL depois da inserção ou remoção de todos os elementos do exemplo 3

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

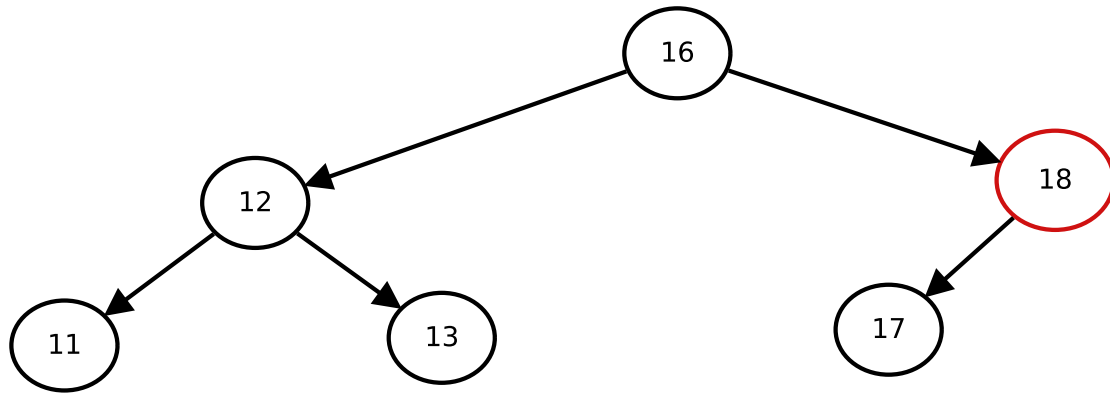


Figura 10: ABB com destaque para a busca do elemento 18.

A Figura 10 exibe a ABB do tipo AVL com destaque para o nó de valor 18, onde será feita uma busca, 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 04