

# AED2 2023 (1s) - AP8 - ÁRVORES BALANCEADAS

---

## 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 <https://judge.unifesp.br/aediiS1A23/>;

## Descrição:

Queremos comparar as operações realizadas na construção de uma Árvore Vermelha e Preta (AVP) com as realizadas para construir uma Árvore AVL. Reaproveitando o código da AP7 (árvores AVL), adicione a ele as operações básicas de um algoritmo de pesquisa em AVP. Seu programa **deve** conter os seguintes procedimentos: (1) inicialização, (2) pesquisa, (3) inserção, (4) cálculo da altura do nó e (5) cálculo da altura-negra (altura máxima contando apenas os nós de cor preta). Para isso, considere:

- A altura de um nó  $x$  em uma AVP é a mesma já utilizada nas atividades anteriores, sobre ABB e AVL. 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;
- A **altura-negra** de um nó em uma AVP é a distância entre o nó  $x$  e o seu descendente mais afastado **contando apenas os nós de cor preta**, incluindo o próprio nó  $x$ , se este for da cor preta.

Lembre-se que, cada inserção em uma árvore AVP 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.

Seu código deve ser capaz de contar, no caso das árvores AVP, o número total de mudanças de cor nos nós da árvore. No caso da AVL, queremos verificar o número total de rotações que ocorrem no balanceamento da árvore.

Considere as seguintes condições:

1. A complexidade de cada procedimento implementado deve ser 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;

4. Note que, ao contrário da AP7, nesta atividade os nós encontrados na árvore AVL **não devem** ser removidos.

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 deverá ser inserido na árvore.

A terceira linha contém um único valor inteiro positivo a ser apenas **pesquisado** na árvore. Esse número pode ou não estar na árvore. Caso o valor não seja encontrado, ele **não** deve ser inserido.

#### **SAÍDA:**

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 para todos os nós pesquisados e encontrados, ou seja, nós que já foram inseridos na árvore, seguido pela altura da esquerda e direita. Cada resultado deverá ser exibido em uma nova linha.

Na penúltima linha de saída deve ser impresso o valor da **altura negra** do nó pesquisado (linha 03 dos dados de entrada). Caso esse valor pesquisado não seja encontrado, deve ser exibido "Valor nao encontrado".

Na última linha da saída devem ser impressos dois números inteiros: o número total de mudanças de cor em nós da árvore AVP, e o número de rotações realizadas na montagem da árvore AVL. Esses números devem considerar a somatória da entrada original e dos elementos adicionados na linha 2 da entrada.

### Exemplos de entrada e saída:

- *input01*:

Entrada	Saída
6 4 3 2 1 -1	2, 2, 1
2 5 -1	1, 1, 1
6	1
	7, 2

Tabela 1: Exemplos de entrada e saída 01

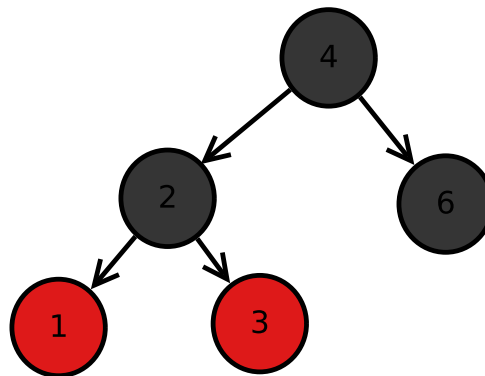


Figura 1: ABB do tipo vermelha e preta depois da inserção de todos elementos de entrada

A Figura 1 exibe a árvore Vermelha e Preta 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  $hd = 1$ .

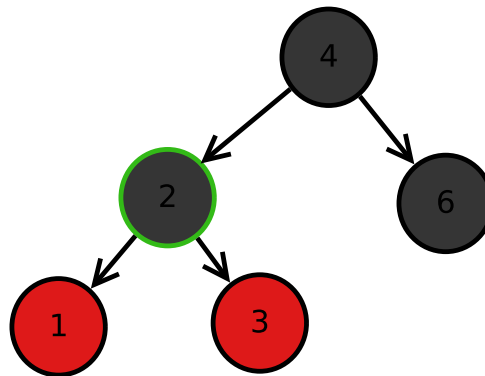


Figura 2: ABB do tipo Vermelha e preta com destaques para o nó 2, com borda verde, a ser pesquisado

Na Figura 2 tem-se a árvore binária do tipo Vermelha e Preta, onde o nó de valor 2, em destaque com borda verde, deverá retornar a altura, a altura à esquerda e à direita,  $h = 1$ ,  $he = 1$  e  $hd = 1$ .

Na Figura 3 tem-se a árvore após a última inserção, referente ao valor 5. Após esta etapa ainda será feita uma busca pelo valor 6 (definido na terceira linha dos dados de entrada), o qual, conforme se pode verificar na mesma Figura 3, tem altura negra 1.

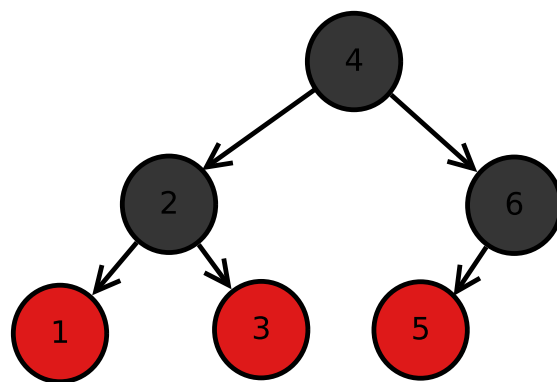


Figura 3: ABB Vermelha e Preta após a inserção do valor 5

- *input02*

Entrada	Saída
6 4 3 2 1 -1	2, 2, 1
2 5 -1	1, 1, 1
5	0
	7, 2

Tabela 2: Exemplos de entrada e saída 02

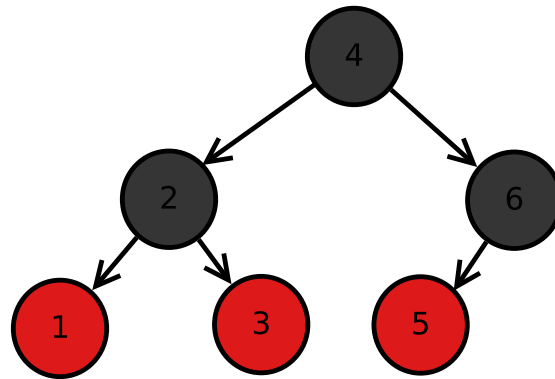


Figura 4: ABB do tipo vermelha e preta final

A Figura 4 exibe a árvore Vermelha e Preta referente a entrada de dados constante da linha 01, igual ao *input01*, porém agora a busca é para o valor 5 que retornará 0 de altura negra.

- *input03*

Entrada	Saída
6 4 3 2 1 5 7 8 -1	3, 2, 3
6 4 9 11 10 8 4 -1	2, 1, 2
6	3, 2, 3
	2, 1, 2
	2, 2, 1
	2
	19, 5

Tabela 3: Exemplos de entrada e saída 03

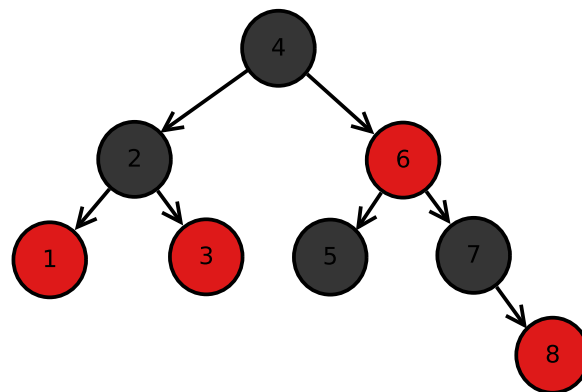


Figura 5: Árvore inicial do input3, após a inserção dos valores da primeira linha

A Figura 5 exibe a árvore Vermelha e Preta referente a entrada de dados constante da linha 01 do primeiro exemplo, onde a altura a partir do nó raiz é  $h = 3$ , com altura da esquerda  $he = 2$  e da direita  $hd = 3$ .

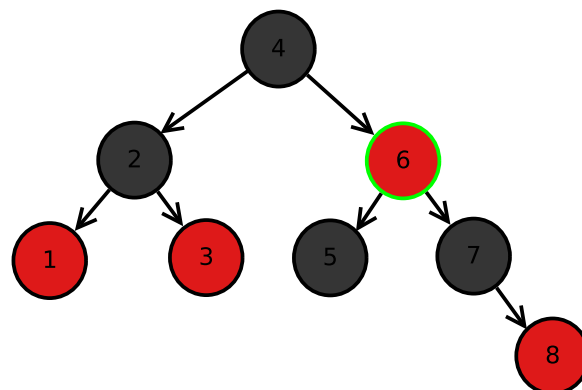


Figura 6: AVP com destaque para elemento de valor 6 buscado na árvore.

A Figura 6 destaca o valor 6, nó com borda verde, primeiro valor a tentar ser inserido, porém como o valor 6 já está inserido, deve-se exibir a altura do nó 6, altura da esquerda e da direita, ou seja,  $h = 2$ , com altura da esquerda  $he = 1$  e da direita  $hd = 2$ .

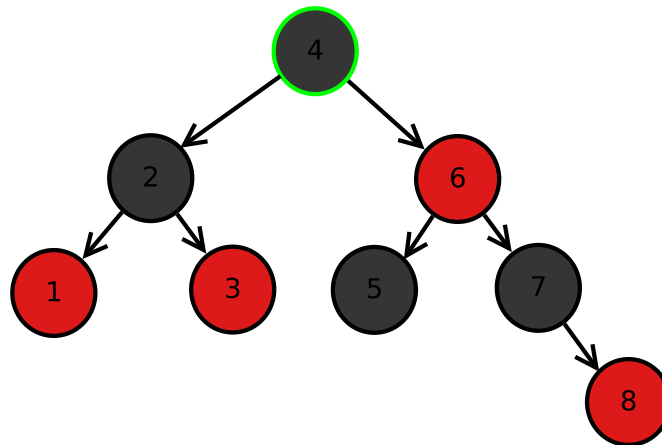


Figura 7: AVP com destaque para elemento de valor 4 buscado na árvore.

A Figura 7 refere-se a tentativa de buscar ou inserir o próximo valor da segunda linha de entrada, referente ao valor 4. Uma vez que este valor também já encontra-se na árvore, deve-se exibir os dados referente a altura do citado nó 4, ou seja,  $h = 3$ ,  $he = 2$  e  $hd = 3$ .

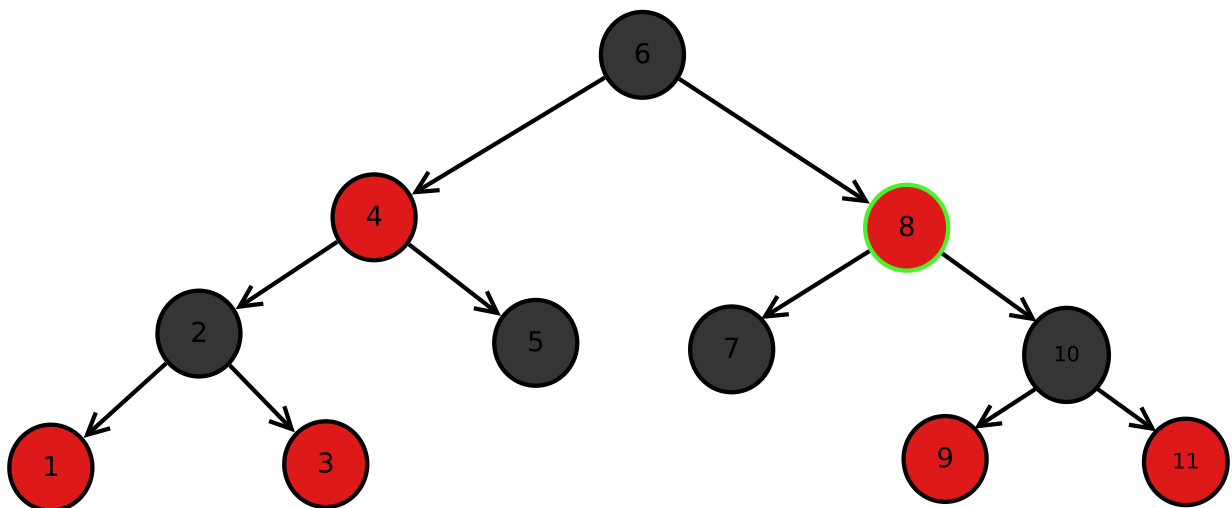


Figura 8: AVP após a inserção dos elementos constantes da segunda linha dos dados de entrada do input3.

A Figura 8 retrata como a AVP ficou após a inserção de todos os valores constantes da segunda linha dos dados de entrada. Destaca-se na mesma figura o valor 8, que já estava inserido desde a primeira linha, sendo necessário portanto retornar seus dados de altura, ou seja,  $h = 2$ ,  $he = 1$  e  $hd = 2$ .

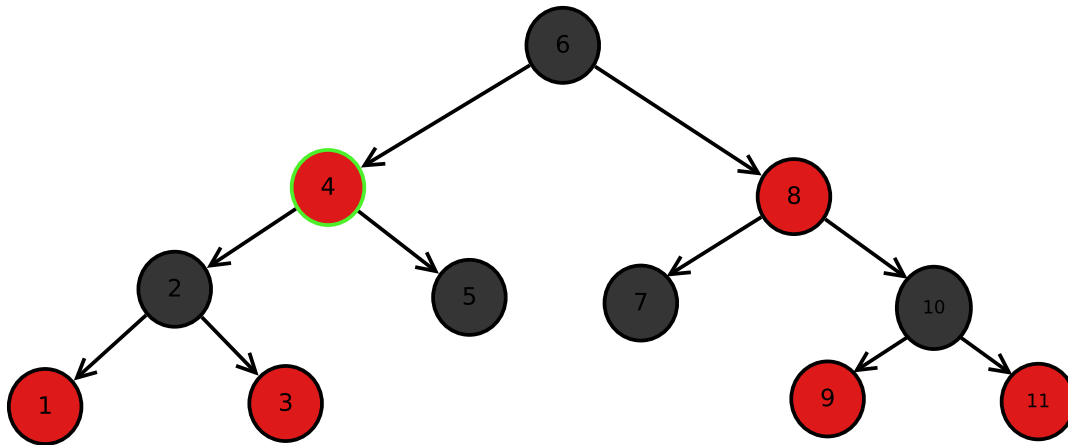


Figura 9: AVP final com destaque para a busca bem sucedida pelo nó de valor 4, já existente.

A Figura 9 destaca o valor 4, último valor da segunda linha dos dados de entrada. Como o valor 4 já está inserido, então é necessário retornar dados de sua altura, ou seja,  $h = 2$ ,  $he = 2$  e  $hd = 1$ . Após a busca pelo valor 4, deve ser efetuada uma busca pelo valor 20, conforme definido na última linha dos dados de entrada.



- *input04*

Entrada	Saída
6 4 3 2 1 5 7 8 -1	3, 2, 3
6 4 9 11 10 8 4 -1	2, 1, 2
20	3, 2, 3
	2, 1, 2
	2, 2, 1
	Valor nao encontrado
	19, 5

Tabela 4: Exemplos de entrada e saída 03

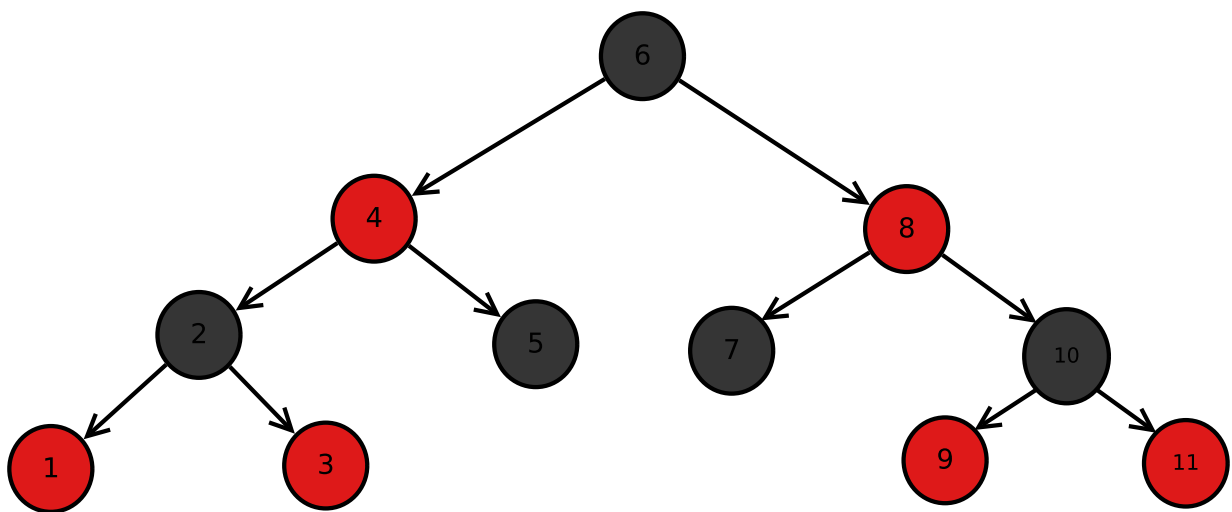


Figura 10: AVP após a inserção dos elementos constantes da segunda linha dos dados de entrada do *input4*.

A Figura 10 representa a árvore final após as inserções das linhas 01 e 02, igual ao *input03*. Porém a busca é para o valor 20. Como este valor não existe na AVP, deve-se retornar a mensagem: “Valor nao encontrado”.