

AED2 2021 - TAREFA 10 - ÁRVORE VERMELHO E PRETA

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++, Java, Python 2 ou 3.

Descrição: Implemente as operações básicas de um algoritmo de pesquisa para Árvores Vermelho e Preto (AVP) conforme as especificações abaixo. Escreva um programa que contenha os procedimentos a seguir: (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). A altura de um nó (procedimento 4) x em uma AVP 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 (mesmo procedimento já utilizado nas atividades anteriores sobre árvore). A altura-negra de um nó (procedimento 5) 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.

Deve-se lembrar que em uma AVP a inserção já é balanceada, ou seja, a cada inserção deve-se checar as regras estabelecidas para AVPs e, se necessário, executa-se transformações na estrutura da árvore de forma a mantê-la balanceada. Obs: As entradas de dados tem a mesma estrutura das atividades anteriores.

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 Vermelho e Preto.

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 o valor seja encontrado na AVP, deverá ser exibido como saída a altura desse nó.

A terceira linha representa um único valor inteiro positivo a ser apenas pesquisado na árvore, o qual pode ou não existir. Caso o valor não seja encontrado, ele NÃO deve ser inserido.

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 Vermelha e Preta a partir do seu nó raiz, seguido da altura da esquerda e da direita do nó raiz.

Na linha seguinte 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 última linha de saída deve ser impresso o valor da **altura negra** do nó pesquisado (linha 03 dos dados de entrada), a qual se refere a altura máxima contando apenas os nós de cor preta, a partir do nó pesquisado. Importante ressaltar nesse caso que o cálculo da altura DEVE INCLUIR o próprio nó pesquisado, caso ele seja de cor preta. 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 |
| 2 5 -1 | 1, 1, 1 |
| 6 | 1 |

Tabela 1: Exemplos de entrada e saída 01

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 $hd = 1$.

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 $hd = 1$.

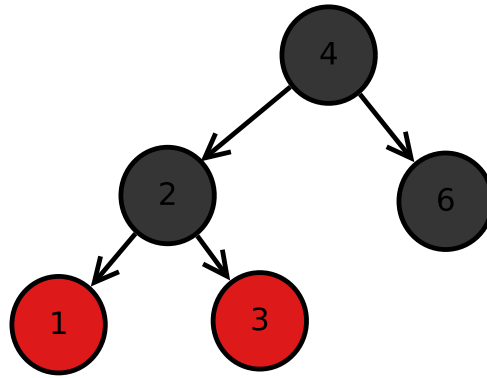


Figura 1: ABB do tipo vermelha e preta depois da inserção de todos elementos de entrada

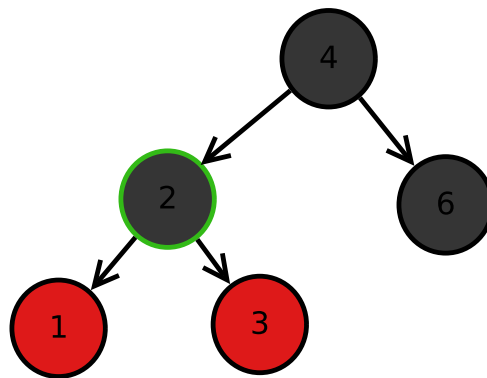


Figura 2: ABB do tipo Vermelha e preta com destaques para o nó 2, com borda verde, a ser pesquisado

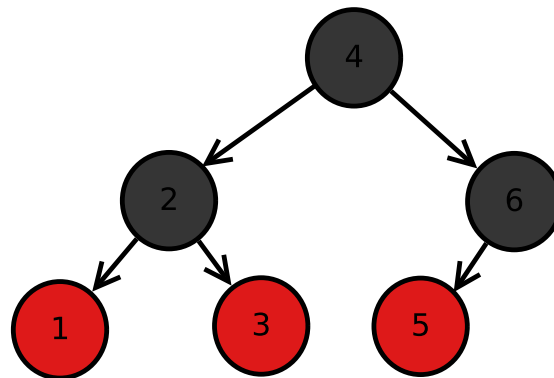


Figura 3: ABB Vermelha e Preta após a inserção do valor 5

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.

- *input02*

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

Tabela 2: Exemplos de entrada e saída 02

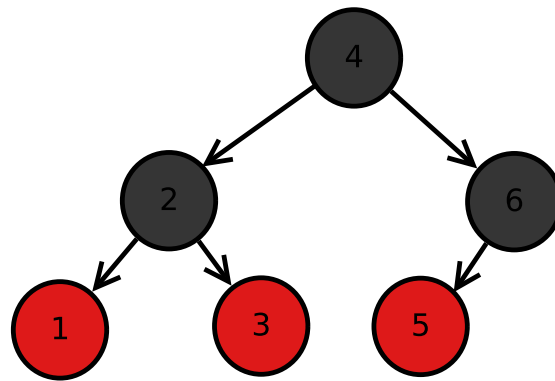


Figura 4: ABB do tipo vermelha e preta final

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 |

Tabela 3: Exemplos de entrada e saída 03

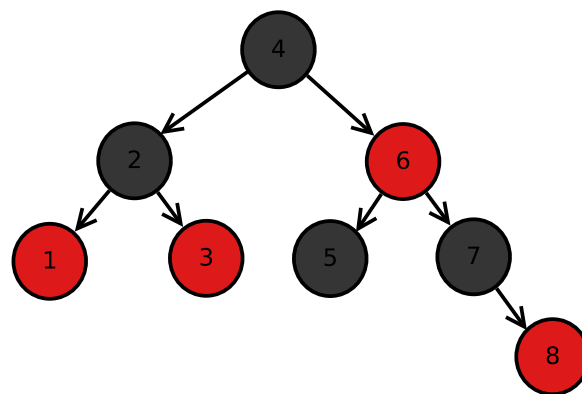


Figura 5: Árvore inicial do input3, após a inserção dos valores da primeira linha

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 $hd = 3$.

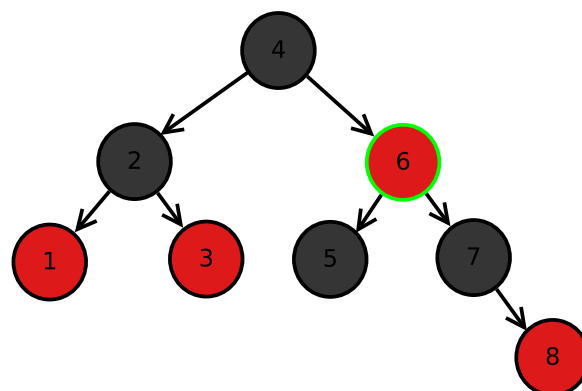


Figura 6: AVP com destaque para elemento de valor 6 buscado na árvore.

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 $hd = 2$.

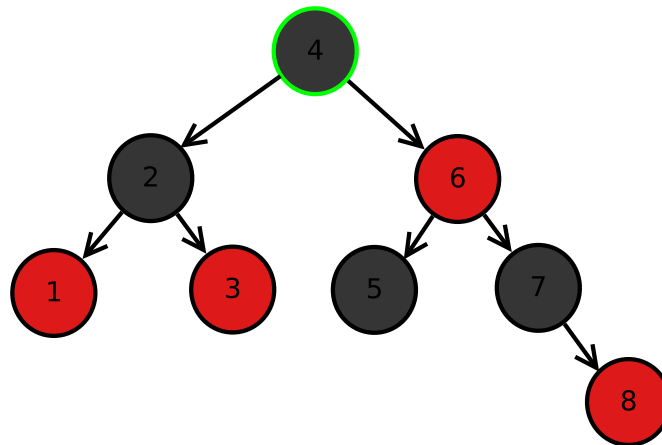


Figura 7: AVP com destaque para elemento de valor 4 buscado na árvore.

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 $hd = 3$.

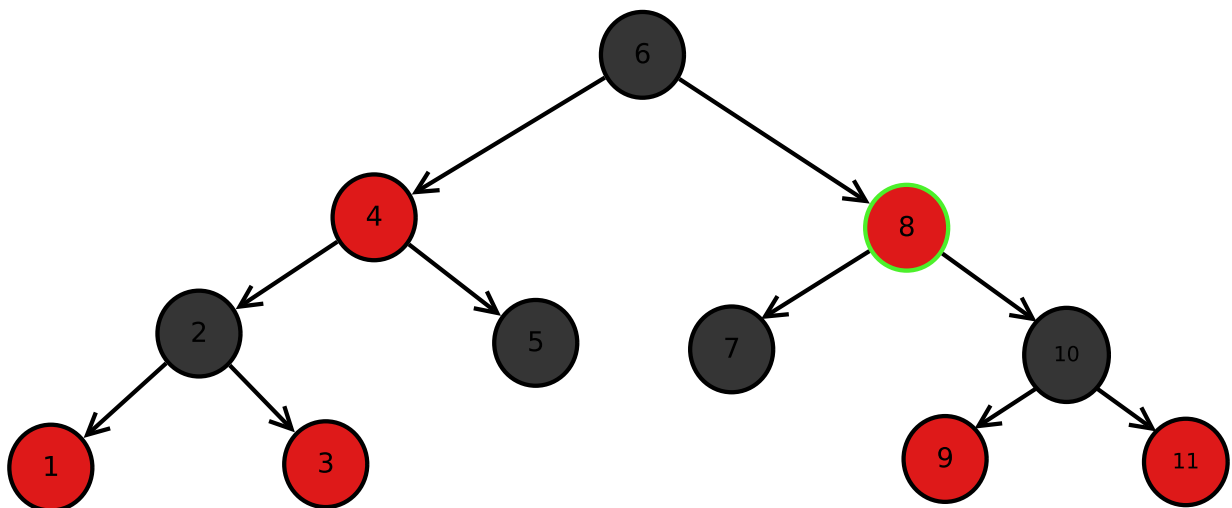


Figura 8: AVP após a inserção dos elementos constantes da segunda linha dos dados de entrada do input3.

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 $hd = 2$.

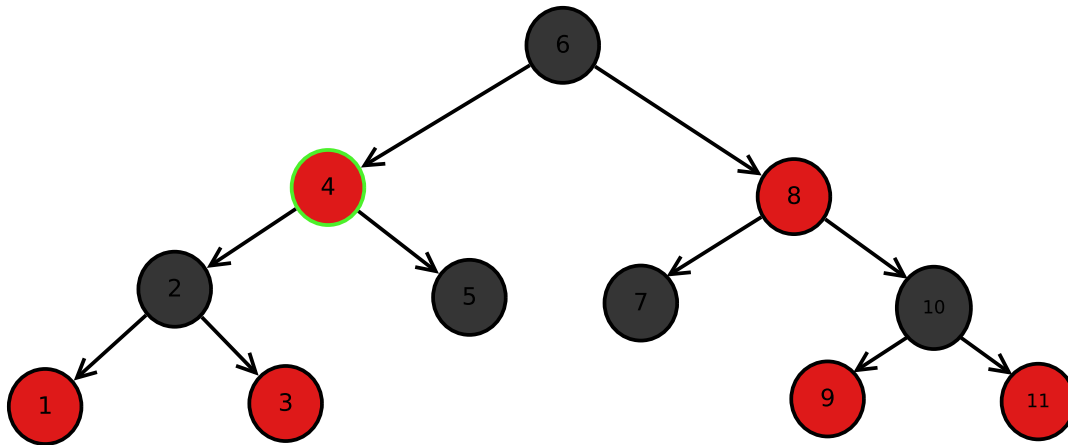


Figura 9: AVP final com destaque para a busca bem sucedida pelo nó de valor 4, já existente.

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 |

Tabela 4: Exemplos de entrada e saída 03

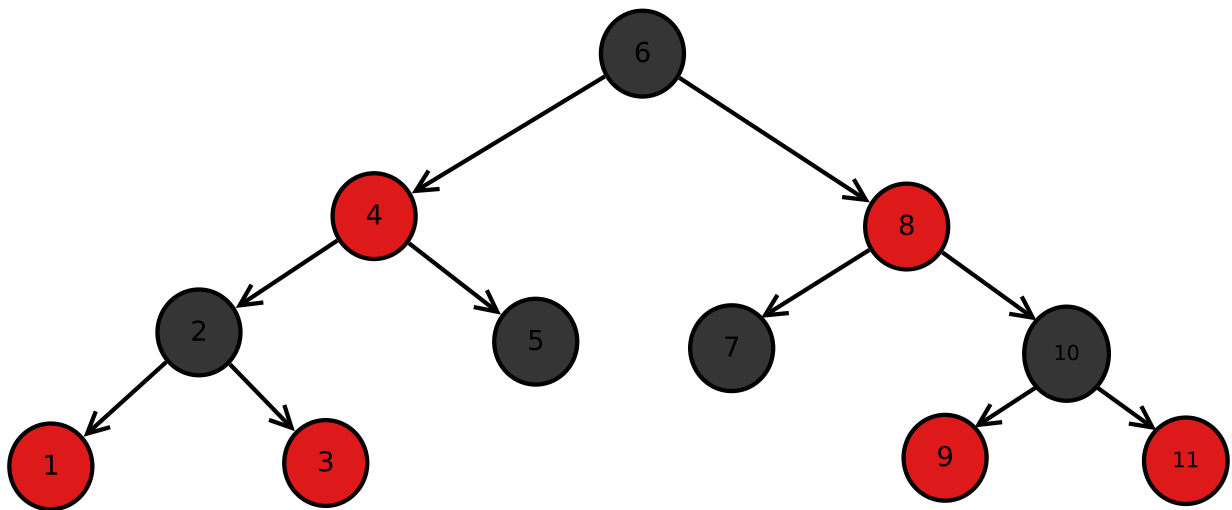


Figura 10: AVP após a inserção dos elementos constantes da segunda linha dos dados de entrada do *input4*.

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".