

# AED2 2025 (1s) - AP06 - ABBs 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 <http://judge.unifesp.br/AEDII202401>;

## Descrição:

Queremos comparar as operações realizadas na construção de uma Árvore Vermelha e Preta (AVP) com as realizadas para construir uma Árvore AVL. Seu programa **deve** conter os seguintes procedimentos para ambos os tipos de árvore binária: (1) inicialização, (2) pesquisa, (3) inserção, (4) cálculo da altura do nó. Para a árvore AVP, deve conter também (5) cálculo da altura-negra (altura máxima contando apenas os nós de cor preta).

Para fins de comparação, você deverá criar uma árvore de cada tipo (1 AVP e 1 AVL). Todas as operações realizadas sobre as entradas deverão ocorrer nas duas árvores de maneira independente, de forma que elas contenham, ao final do programa, os mesmos valores.

Para isso, considere:

- A altura de um nó  $x$  em uma AVP é a mesma já utilizada nas atividades anteriores, sobre 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. Nesse caso consideraremos, apenas para efeito de saída de informações na atividade, que o nó folha aqui descrito, é o nó válido, e não o nó folha nulo;
- 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 AVL requer que se verifique o fator de balanceamento e, se necessário, uma rotação é realizada. Nas árvores AVP deve-se verificar, igualmente a cada inserção, se os critérios de uma AVP estão satisfeitos, caso contrário o balanceamento se dará por trocas de cores e/ou rotações.

Seu código deve ser capaz de contar o total de rotações executadas no balanceamento das árvores (ou seja, a somatória do número de rotações executadas na inserção de todos os elementos da árvore). No caso das árvores AVP, deve ser contado também, independentemente, o número total de mudanças de cor nos nós 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++;
3. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;
4. Note que, ao contrário da atividade anterior, nesta atividade os nós encontrados na árvore **não devem** ser removidos.

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

#### **ENTRADA:**

A entrada consiste de uma lista de números inteiros positivos separados por espaços. Esses números devem ser inseridos nas árvores AVL e AVP independentemente uma da outra. A sequência termina com um número inteiro negativo que não deve ser inserido.

#### **SAÍDA:**

A primeira linha da saída contém a altura máxima da árvore AVL a partir do seu nó raiz seguida da altura da sub-árvore da esquerda e da direita do nó raiz.

A segunda linha deve conter a altura máxima da árvore vermelha e preta a partir do seu nó raiz, seguida da altura da sub-árvore da esquerda e da direita do nó raiz.

A terceira linha deverá conter um único número inteiro: a altura negra da árvore AVP, contada a partir da raiz.

A última linha deverá conter três números inteiros: a quantidade total de mudanças de cor de nós da AVP, a quantidade total de rotações executadas na AVP e, por fim, a quantidade total de rotações executadas na AVL.

### Exemplos de entrada e saída:

- *input01*:

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

Tabela 1: Exemplos de entrada e saída 01

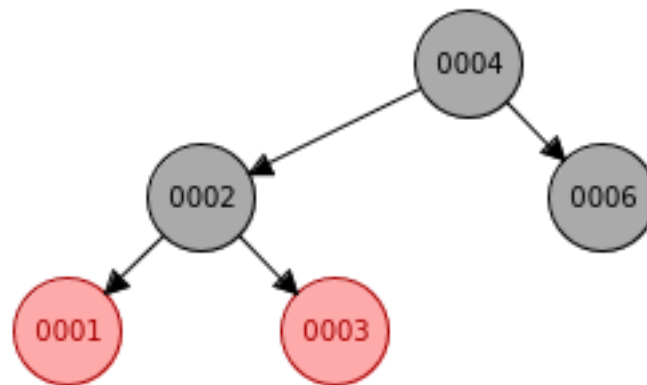


Figura 1: ABB do tipo vermelha e preta depois da inserção de todos elementos de entrada

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  $hd = 1$ . A altura negra a partir da raiz é  $hn = 2$ .

- *input02*

Entrada	Saída
6 4 5 3 2 7 1 -1	3, 3, 2
	3, 3, 2
	2
	10, 3, 2

Tabela 2: Exemplos de entrada e saída 02

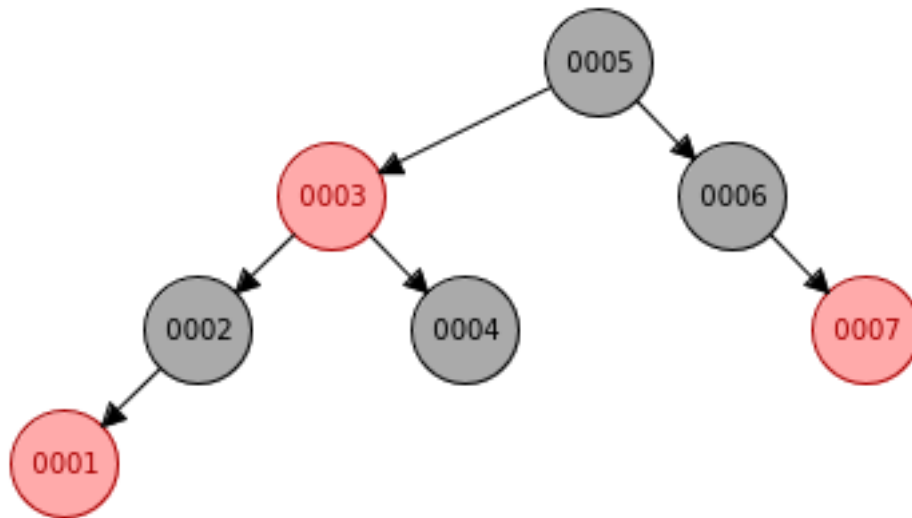


Figura 2: ABB do tipo vermelha e preta final

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

- *input03*

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

Tabela 3: Exemplos de entrada e saída 03

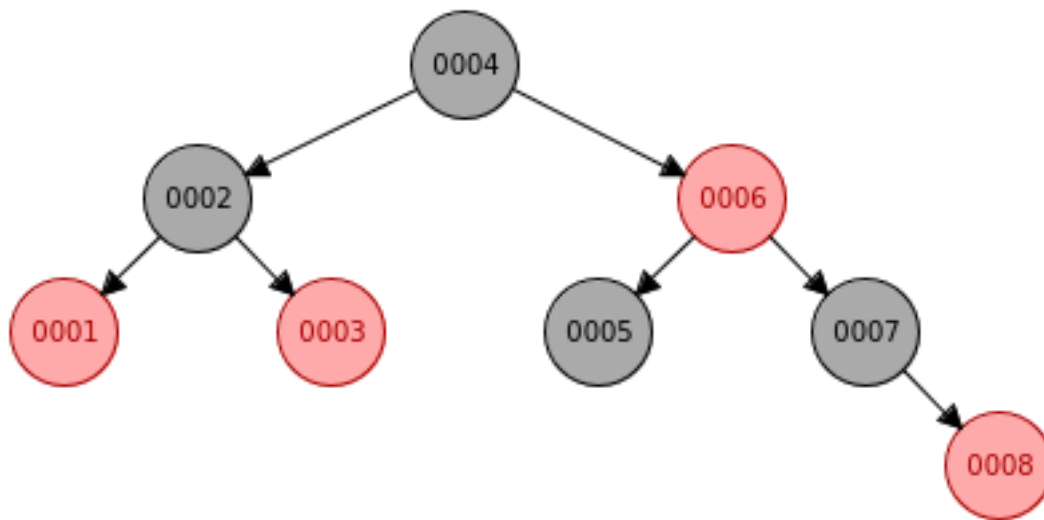


Figura 3: Árvore inicial do input3, após a inserção dos valores da primeira linha

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

- *input04*

Entrada	Saída
6 4 3 2 1 5 7 8 11 9 10 15 13 -1	3, 3, 3
	4, 3, 4
	3
	27, 8, 7

Tabela 4: Exemplos de entrada e saída 04

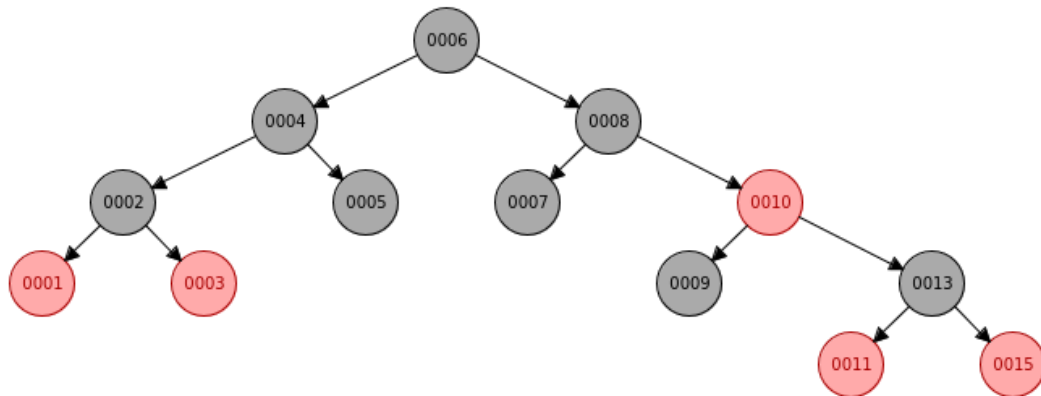


Figura 4: AVP após a inserção dos elementos constantes da primeira linha dos dados de entrada do input4.

A Figura 4 representa a árvore final após as inserções da linha 01 do *input04*, onde a altura a partir do nó raiz é  $h = 4$ , com altura da esquerda  $h_e = 3$  e da direita  $h_d = 4$ . A altura negra a partir da raiz é  $h_n = 3$ .