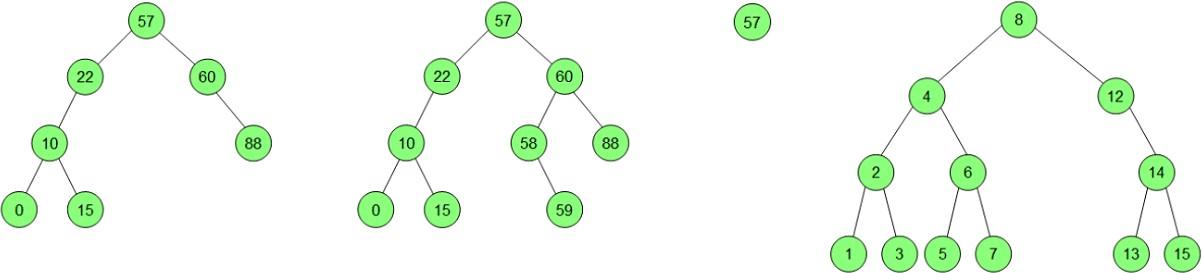
# Exercício – Aula 07 Árvores AVL

**Turma:** Estrutura de Dados II (2023.1)

**Aluno:** Eduardo Henrique de Almeida Izidorio

**matrícula:** 2020000315

# Para cada árvore binária abaixo, diga se é uma árvore AVL (justifique).



| Não, nó 22 desbalanceado | Não, nó 22 desbalanceado | sim, balanceado | Não, nó 12 desbalanceado |
| --- | --- | --- | --- |

1. **Monte a árvore AVL (passo-a-passo) para as seguintes inserções de chaves, indicando a cada passo qual elemento foi inserido ou qual rotação foi realizada:**

**a)** 50, 30, 20, 70, 40, 35, 37, 38, 10, 32, 45, 42, 25, 47, 36

| **ÁRVORE AVL** | **DESCRIÇÃO** |
| --- | --- |
|  | * inserir: 50 * rotação: - |
|  | * inserir: 30 * rotação: - |

|  | * inserir: 20 * rotação: rotação à direita |
| --- | --- |
|  | * inserir:70 * rotação: - |
|  | * inserir: 40 * rotação: - |
|  | * inserir: 35 * rotação: rotação dupla à direita   + 1: direita   + 2: esquerda |

|  | * inserir: 37 * rotação: - |
| --- | --- |
|  | * inserir: 38 * rotação: rotação à esquerda |
|  | * inserir: 10 * rotação: - |
|  | * inserir: 32 * rotação: rotação dupla à esquerda * 1: esquerda * 2: direita |

|  |  |
| --- | --- |
|  | * inserir: 45 * rotação: - |
|  | * inserir: 42 * rotação: rotação dupla à direita   + 1: direita   + 2: esquerda |
|  | * inserir: 25 * rotação: - |
|  | * inserir: 47 * rotação: - |

|  | * inserir: 36 * rotação: - |
| --- | --- |

**b)** 100, 80, 60, 40, 20, 70, 30, 50, 35, 45, 55, 75, 65, 73, 77

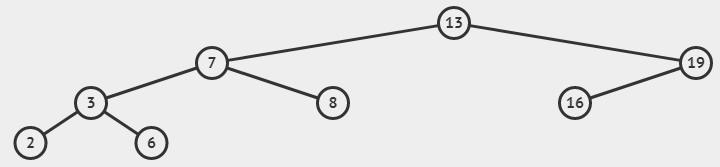
| **ÁRVORE AVL** | **DESCRIÇÃO** |
| --- | --- |
|  | * inserir: 100 * rotação: - |
|  | * inserir: 80 * rotação: - |
|  | * inserir: 60 * rotação: rotação à direita |

|  | * inserir: 40 * rotação: - |
| --- | --- |
|  | * inserir: 20 * rotação: rotação à direita |
|  | * inserir: 70 * rotação: rotação dupla à esquerda * 1: esquerda * 2: direita |
|  | * inserir: 30 * rotação: rotação dupla à esquerda * 1: esquerda * 2: direita |

|  |  |
| --- | --- |
|  | * inserir: 50 * rotação: - |
|  | * inserir: 35 * rotação: - |
|  | * inserir: 45 * rotação: rotação à esquerda |
|  | * inserir: 55 * rotação: - |

|  | * inserir: 75 * rotação: - |
| --- | --- |
|  | * inserir: 65 * rotação: - |
|  | * inserir: 73 * rotação: rotação dupla à esquerda * 1: esquerda * 2: direita |
|  | * inserir: 77 * rotação: - |

1. **Dadas as seguintes chaves M, G, B, H, S, P, F, C como entrada (nesta ordem), desenhe a respectiva árvore AVL (balanceando-a quando for necessário).**

* M - 13
* G - 7
* B - 2
* H - 8
* S - 19
* P - 16
* F - 6
* C - 3

1. **Monte a árvore AVL (passo-a-passo) para as seguintes inserções de chaves 41, 38, 31, 12, 19, 8, 27, 49 (nesta ordem),**

* Indicando a cada passo qual elemento foi inserido,
* O grau de balanceamento de cada nó e
* Qual rotação foi realizada.

1. **Implemente a árvore AVL. Para um bom entendimento tente utilizar o arquivo bst.py (como fizemos em sala) e implemente as funções**

# Funções Auxiliares

**def getHeight(root):**

# FB = H(T.left) - H(T.right)

**def getBalance(root):**

# ROTAÇÕES

**def leftRotate(z): def rightRotate(z):**

# Função de inserção

**def insertNode(root, value):**

# Faz a inserção de maneira recursiva

# Atualiza altura (pois um filho foi adicionado) # Verificar o Fator de Balanceamento

# Chama as rotações para os casos específicos # Função de remoção

**def deleteNode(root, value)**