



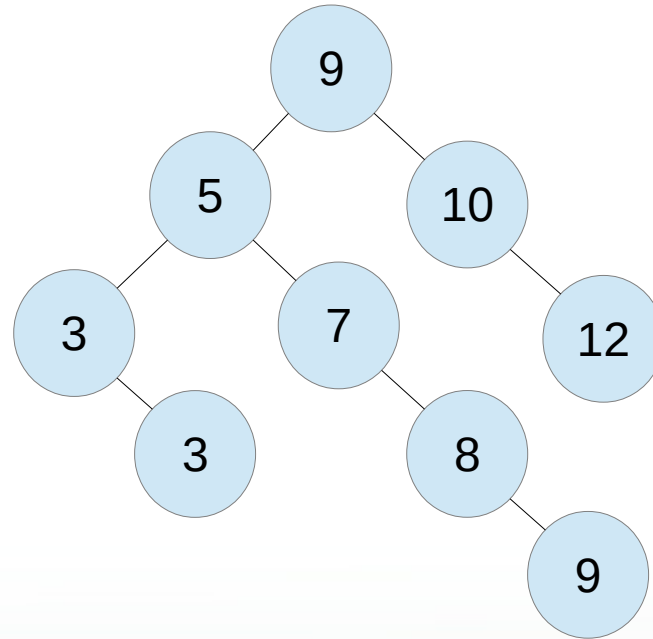
# Exercícios Programação II - Árvores

- Questões:

1. Monte as árvores (graficamente), a partir da entrada a seguir:
  - a)  $E1 = 9 - 5 - 3 - 7 - 4 - 10 - 8 - 12 - 9$
  - b)  $E2 = 20 - 22 - 3 - 14 - 15 - 6 - 33 - 23 - 0$
  - c)  $E3 = 15 - 12 - 5 - 14 - 15 - 3 - 3 - 2 - 8$
2. Quais são as impressões para os Algoritmos a seguir a partir das árvores geradas no exercício 1:
  - a) inOrder
  - b) preOrder
  - c) posOrder
3. Faça uma pesquisa e crie um resumo sobre árvores AVL e, explique o que é Rotacionamento.

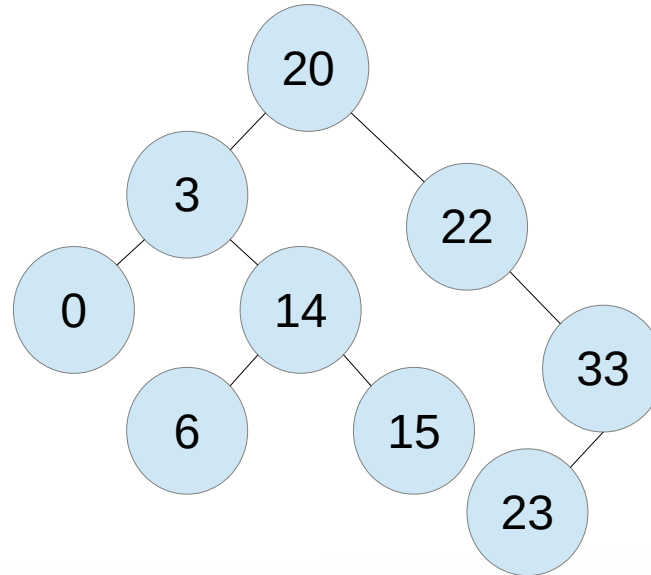
## Respostas 1.a.)

$$E1 = 9 - 5 - 3 - 7 - 4 - 10 - 8 - 12 - 9$$



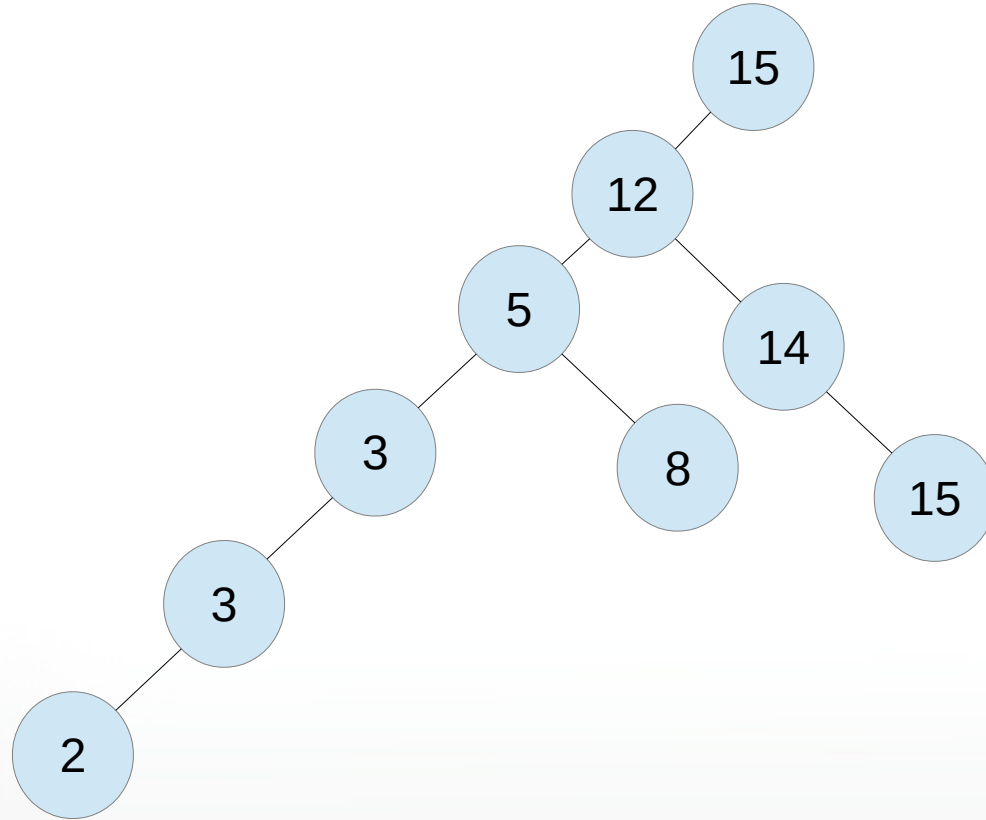
## Respostas 1.b.)

$$E2 = 20 - 22 - 3 - 14 - 15 - 6 - 33 - 23 - 0$$



## Respostas 1.c.)

$$E3 = 15 - 12 - 5 - 14 - 15 - 3 - 3 - 2 - 8$$



# Respostas 2.)

a) inOrder:

- i.  $E1 = 3 - 4 - 5 - 7 - 8 - 9 - 9 - 10 - 12$
- ii.  $E2 = 0 - 3 - 6 - 14 - 15 - 20 - 22 - 23 - 33$
- iii.  $E3 = 2 - 3 - 3 - 5 - 8 - 12 - 14 - 15 - 15$

b) preOrder:

- i.  $E1 = 9 - 5 - 3 - 4 - 7 - 8 - 9 - 10 - 12$
- ii.  $E2 = 20 - 3 - 0 - 14 - 6 - 15 - 22 - 33 - 23$
- iii.  $E3 = 15 - 12 - 5 - 3 - 3 - 2 - 8 - 14 - 15$

c) posOrder

- i.  $E1 = 4 - 3 - 9 - 8 - 7 - 5 - 12 - 10 - 9$
- ii.  $E2 = 0 - 6 - 15 - 14 - 3 - 23 - 33 - 22 - 20$
- iii.  $E3 = 2 - 3 - 3 - 8 - 5 - 15 - 14 - 12 - 15$



# Respostas 3.)

- **Árvores AVL:** é uma árvore binária de busca balanceada, ou seja, uma árvore balanceada (*árvore completa*), são as árvores que minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas. Contudo, para garantir essa propriedade em aplicações dinâmicas, é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (*inclusão e exclusão*), para ser alcançado um custo de algoritmo com o tempo de pesquisa a  $O(\log n)$ . As operações de busca, inserção e remoção de elementos possuem complexidade  $O(\log n)$  (*no qual  $n$  é o número de elementos da árvore*), que são aplicados a árvore de busca binária. O nome **AVL**, vem de seus criadores soviéticos *Adelson Velsky* e *Landis*, e sua primeira referência encontra-se no documento "*Algoritmos para organização da informação*" de 1962. Nessa estrutura de dados cada elemento é chamado de *nó*. Cada nó armazena uma chave e dois ponteiros, uma para a subárvore esquerda e outro para subárvore direita.

Fonte: Wikipédia. Disponível via URL: <[https://pt.wikipedia.org/Árvore\\_AVL](https://pt.wikipedia.org/Árvore_AVL)>. Acesso em 18 de Junho de 2019.



# Respostas 3.)

- **Árvores AVL - Rotação:** A operação básica em uma árvore AVL, geralmente envolve os mesmos algoritmos de uma árvore binária desbalanceada. *A rotação na árvore AVL ocorre de vido ao seu desbalanceamento*, uma rotação simples ocorre quando um nó está desbalanceado, e seu filho estiver no mesmo sentido da inclinação, formando uma linha reta. Uma rotação dupla ocorre quando um nó estiver desbalanceado e seu filho estiver inclinado no sentido inverso ao pai, formando um *“joelho”*. Para garantirmos as propriedades da árvore AVL, rotações devem ser feitas conforme necessário após operações de remoção ou inserção. São conhecido 4 tipo de rotações: *rotação simples à direita, rotação à esquerda, rotação dupla à direita e rotação dupla à esquerda*.

Fonte: Wikipédia. Disponível via URL: <[https://pt.wikipedia.org/Árvore\\_AVL](https://pt.wikipedia.org/Árvore_AVL)>.  
Acesso em 18 de Junho de 2019.