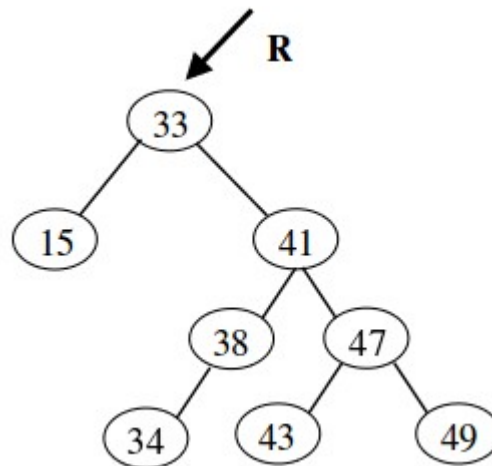


**Lista de Exercícios 5 – Árvores**

Não é necessário entregar esta lista de exercícios.

1) Implemente o TAD ABB.

2) descreva a ordem de visita em ordem, pré-ordem e pós-ordem da árvore abaixo.



3) Considere as seguintes aplicações de árvores:

- Organograma (diagrama hierárquico de uma organização);
- NUSP de alunos e funcionários;
- Árvore genealógica;
- Sistema de gerenciamento de funcionários, indexado por CPF.

Quais delas podem ser modeladas como uma árvore AVL e quais não podem? Explique.

4) Insira em uma árvore AVL os itens com as chaves apresentadas (na ordem em que aparecem). Desenhe a árvore resultante da inserção, sendo que uma nova árvore deve ser desenhada quando houver uma rotação. Indique qual a rotação que foi executada. Entregue uma imagem jpg.

- a) 30, 40, 24, 58, 48, 26, 11, 13, 14
- b) 20, 15, 25, 10, 30, 24, 17, 12, 5, 3
- c) 40, 30, 50, 45, 55, 52
- d) 20, 15, 25, 12, 17, 24, 30, 10, 14, 13
- e) 20, 15, 25, 12, 17, 30, 26

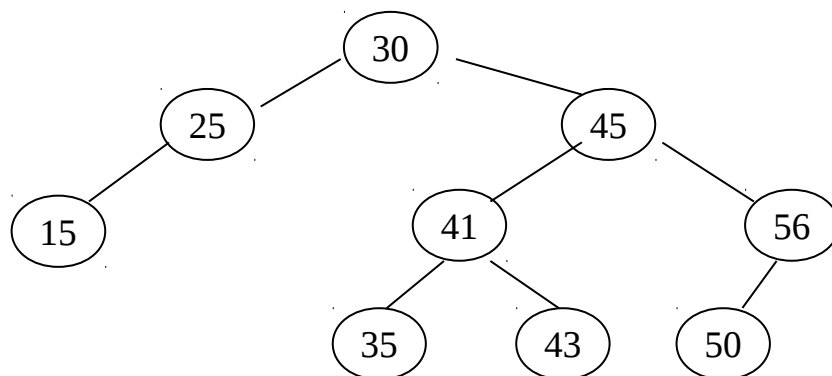
5) Escreva em C uma função não recursiva para determinar a altura de uma **árvore AVL**, visitando o menor número de nós possível.

6) Implemente a busca em largura em uma árvore binária.

7) Qual a diferença de uma ABB para uma AVL?

8) Uma ABB é estritamente binária se todos os nós da árvore tem 2 filhos. Implemente uma função que verifica se uma ABB é estritamente binária.

9) Considere a árvore AVL a seguir:



Realize, na árvore acima, as inserções das seguintes chaves 49, 60, 65, e em seguida a remoção das chaves 45 e 41, escolhendo necessariamente imediatamente precedente para a posição da chave removida. Mostre todas as rotações e o formato da árvore após cada operação.

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

