

Questão 1 (Valor 2.0 pontos) Faça um pseudocódigo que retorne a quantidade de folhas de uma árvore binária.

algoritmo "Quantidade de folhas"

funcao qtdFolhas (No raiz)

inicio

se raiz != nulo então

qtdFolhas(raiz.getEsquerda)

qtdFolhas(raiz.getDireita)

se raiz.getDireita == nulo e raiz.getEsquerda == nulo então

retorne 1

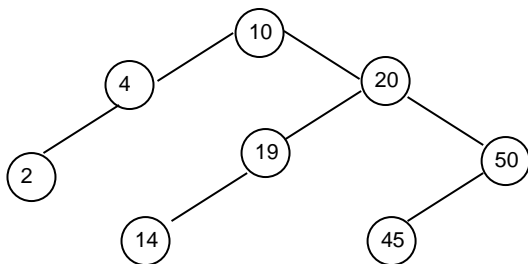
fimse

retorne qtdFolhas(raiz.getDireita) + qtdFolhas(raiz.getEsquerda)

fimse

fimfuncao

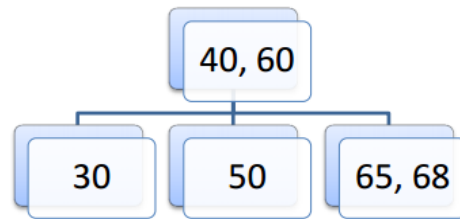
Questão 2 (Valor 1.0 ponto) Apresente os percursos: (a) **pós-ordem** e (b) **pré-ordem**, para a seguinte árvore:



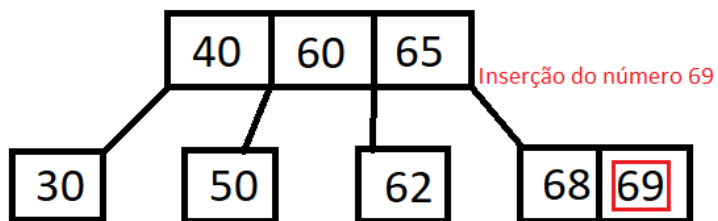
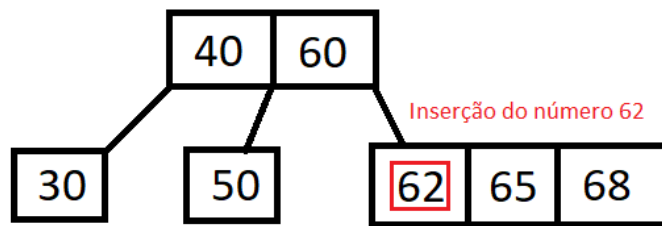
a) pós-ordem: 2, 4, 14, 19, 45, 50, 20, 10

b) pré-ordem: 10, 4, 2, 20, 19, 14, 50, 45

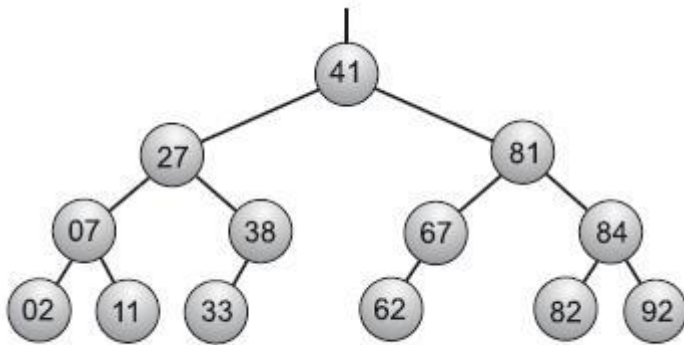
Questão 3 (Valor 1.0 ponto) Dada a árvore 2-3-4 ($t = 2$) abaixo, faça a inserção do



valor 62 e 69.

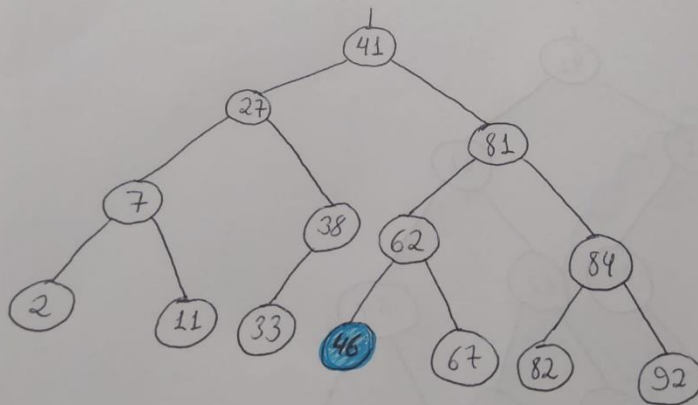


Questão 4 (Valor 2.0 pontos) Dada a árvore AVL abaixo:

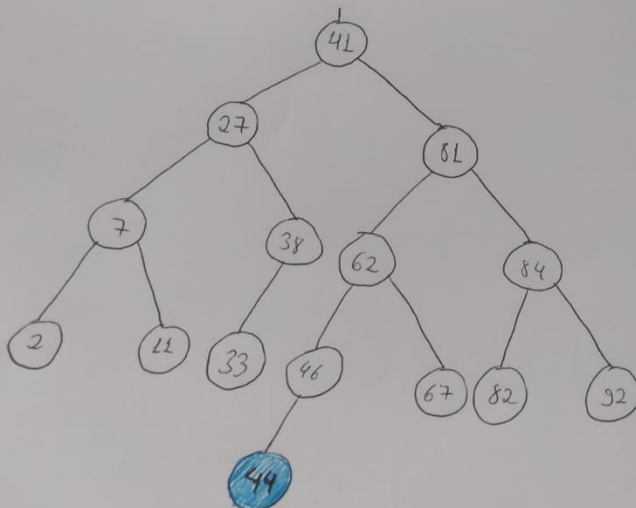


- a) Acrescente os valores 46 e 44 (faça rotações, caso necessário)
- b) A partir da árvore original, remova o valor 81.

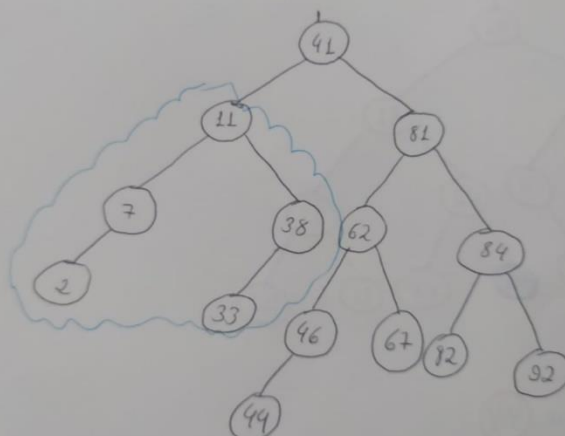
Inserção do número 46



Inserção do número 44



Remoção do número 27



Questão 5 (Valor 2.0 pontos) Suponha que um mapa tenha dois campos, o ID de um produto e quantidade unidades existem em estoque. Suponha, ainda, que o mapa foi implementado através de uma lista simplesmente encadeada. Faça um **método** que entre com o ID e a quantidade de elementos a serem incluídos no estoque; o método deve fazer a atualização da quantidade de um produto, caso ele já exista, ou a inserção deste produto, se não encontrado. Suponha que o método de inserção na lista já tenha sido implementado.

```
bool insereEstoque(int ID, int qtd) {
    NodeMapa *aux;
    aux = inicio;
    while (aux != NULL) {
        if(aux->ID==ID) {
            if(aux->qtd>1) {
                aux->qtd++;
                return true;
            }
            else {
                insere(ID, qtd);
                return true;
            }
        }
        aux = aux -> prox;
    }
    return false
}
```

Questão 6 (Valor 2.0 pontos). Para os dados de entrada: 770, 761, 333, 222, 110, 100 e hash_size = 13.

a) (1.0) Qual o número de colisões para a função: $(n/100 + ((n\%100)\%10))\%hash_size$, onde n é o número inteiro a ser inserido.

b) (1.0) Apresente o vetor com os valores inseridos, através do *Probing linear*.

$$F(770) = 1$$

$$F(761) = 1$$

$$F(333) = 9$$

$$F(222) = 6$$

$$F(110) = 2$$

$$F(100) = 1$$

2 colunas

0	
1	770
2	761
3	110
4	100
5	
6	222
7	
8	
9	333
10	
11	
12	