

1. Desenhe a árvore binária de pesquisa resultante da inserção dos seguintes elementos (nesta ordem): 20, 25, 10, 5, 12, 22, 21, 11, 18, 7, .

a) Escreva a sequência em que os nós seriam percorridos de acordo com as ordens: pré ordem, simétrica e pós-ordem.

b) Desenhe a árvore após a remoção consecutiva dos seguintes nós:

b.1) 10 c.2) 25 b.3) 5

2. Escreva um algoritmo que retorna o número de nós em uma árvore binária que tem grau 1 (um único filho).

int abCalNumNosGrau1 (TNode \*t)

3. Uma árvore t1 é maior que uma árvore t2 se elas são topologicamente iguais e para todos os nós nas posições correspondentes o valor armazenado no nó de t1 é maior que o valor armazenado no nó de t2. Usando o conceito acima, escreva um algoritmo que recebe o nó raiz de duas árvores binárias t1 e t2 e retorna 1 se t1 for maior que t2, 0 de outro modo. Não pode usar memória auxiliar (por exemplo converter a árvore para um vetor.).

int ComparaArvores ( TNode \*t1, TNode \*t2, int (\*getvalue) ( void \*));

OBS: a função getvalue retorna o valor inteiro que está armazenado no nó da árvore.

4. Faça um algoritmo que recebe a raiz de uma árvore binária de pesquisa e remove o elemento de maior valor na árvore

TNode \*abpRemoveMaior(TNode \*t, int (cmp)(void \*, void \*)).

t->r = null