1. Percorra a árvore em pré-ordem, em-ordem e pos-ordem.

a) Pré-ordem:

R: 10, 6, 5, 3, 4, 9, 20, 50, 45

b) Em-ordem:

R: 3, 4, 5, 6. 9, 10 20, 45 50

c) Pós-ordem:

R: 4, 3, 5, 9, 6, 20, 45, 50, 10

10

20

50

45

3

6

9

5

4

1. A partir do código para imprimir uma árvore binária de busca, de maneira recursiva, escreva o código para imprimir de maneira decrescente).

void AVL\_tree::printrecursive(AVL\_node \*r)

{

if(r!=NULL){

cout<<r->key;

printrecursive(r->left);

printrecursive(r->right);

}

}

void AVL\_tree::printrecursive(AVL\_node \*r)

{

if(r!=NULL){

printrecursive(r->right);

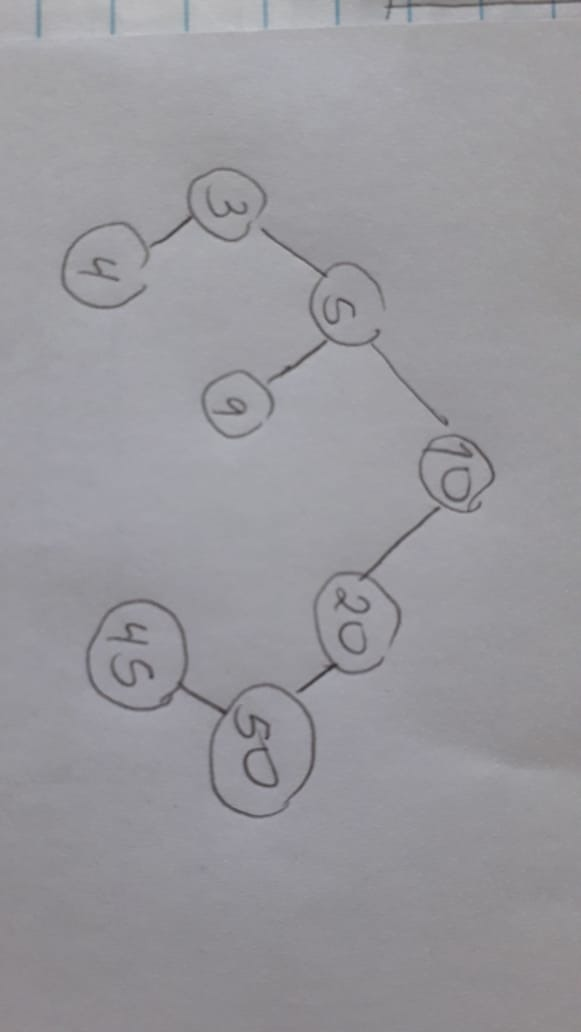
printrecursive(r->left);

cout<<r->key;

}

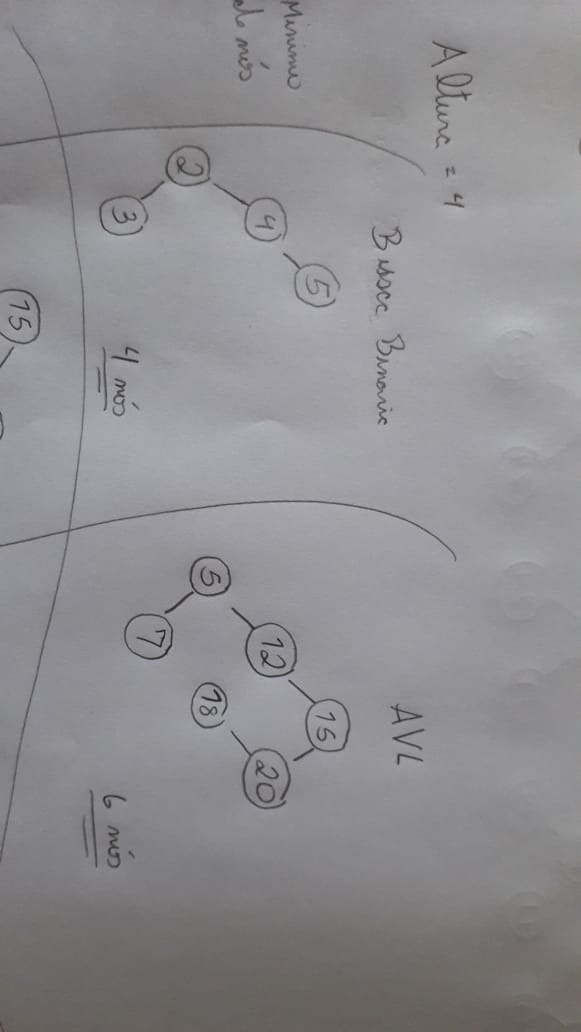
}

1. Construa uma árvore AVL, de maneira que ela sempre fique balanceada (isto é, se a inserção de um elemento causar o desbalanceamento da árvore, é necessário balanceá-la novamente). Apresente a árvore passo-a-passo. Os dados são (nesta ordem): 50, 40, 30, 13, 12, 11
2. Dada a árvore do exercício 1, remova o elemento 6.



1. Considere uma árvore de busca binária e uma árvore AVL de altura 4. Mostre como seria a estrutura destas duas árvores com o número mínimo de nós e com o número máximo de nós.

Minimo de nós:



O Máximo de nós é o mesmo para as 2 arvores.

Máximo de nós:

