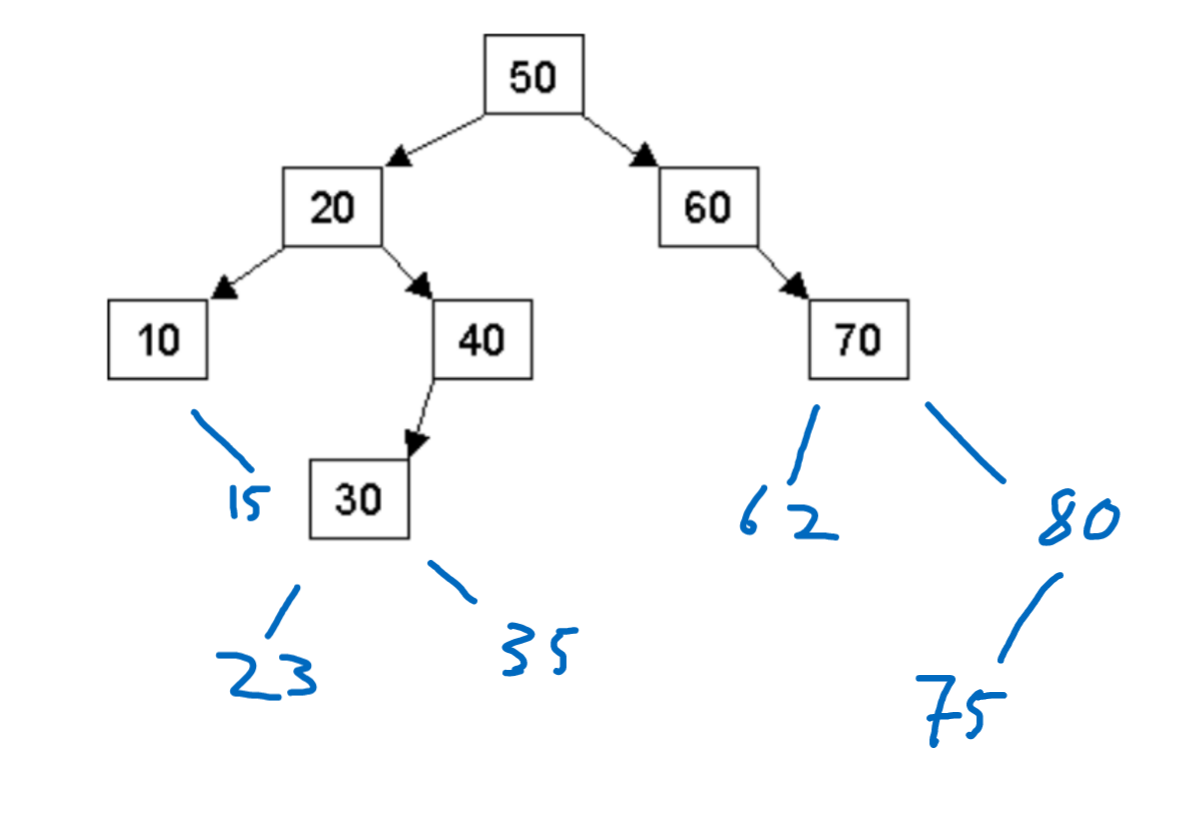
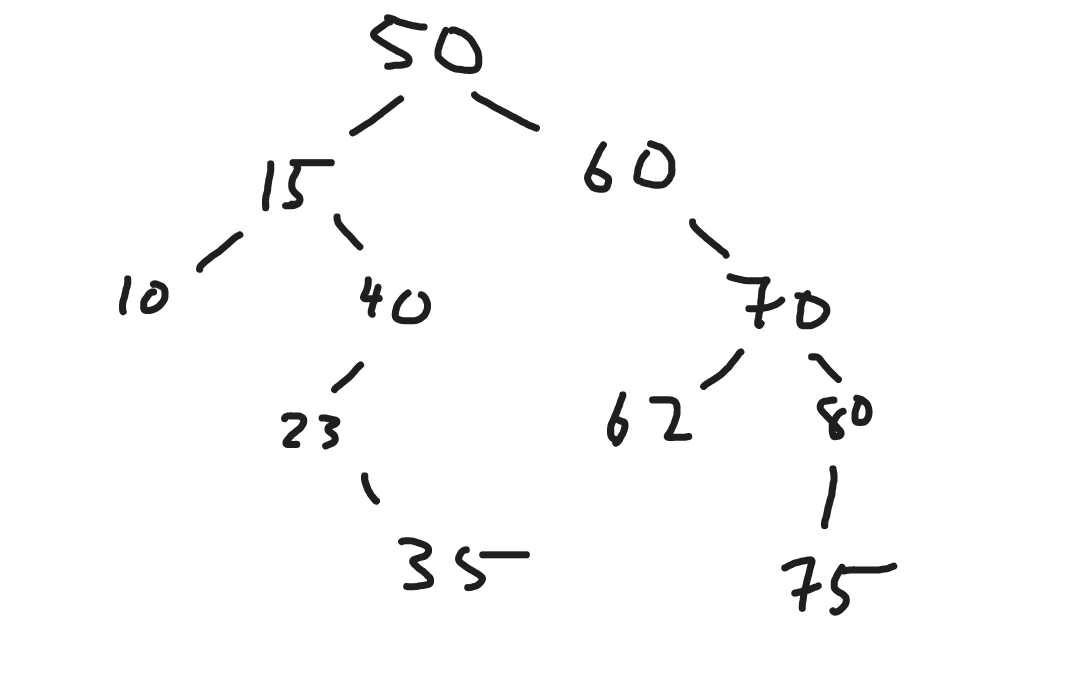
* 1. 
  2. Inorder: 10 15 20 23 30 35 40 50 60 62 70 75 80

Preorder: 50 20 10 15 40 30 23 35 60 70 62 80 75

Postorder: 15 10 23 35 30 40 20 62 75 80 70 60 50

* 1. 

Struct Node {

Node (int val, Node\* daddy) {

left = right = nullptr;

parent = daddy;

value = val;

}

Node \*left, \*right, \*parent;

int value;

}

void helper(Node \*cur, int value, Node \*parent) {

if cur == nullptr

Make new node, set value

left and right = nullptr

set its parent to parent node

Return

Otherwise if data is less than data at cur,

help(cur->left, value, cur)

Else

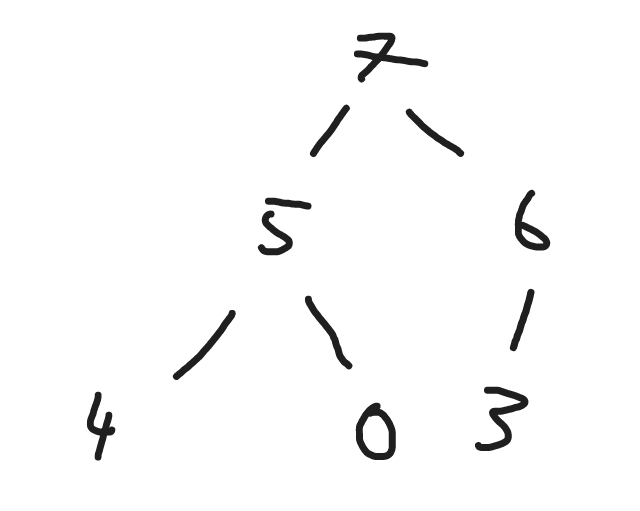
help(cur->right, value, cur)

}

void insert(int value, Node \*start) {

helper(start, value, nullptr);

}

* 1. 
  2. [7, 5, 6, 4, 0, 3]
  3. [6, 5, 3, 4, 0]
  4. O(C + S)
  5. O(logC + S)
  6. O(logC + logS)
  7. O(logS)
  8. O(1)
  9. O(logC + S)
  10. O(SlogS)
  11. O(ClogS)