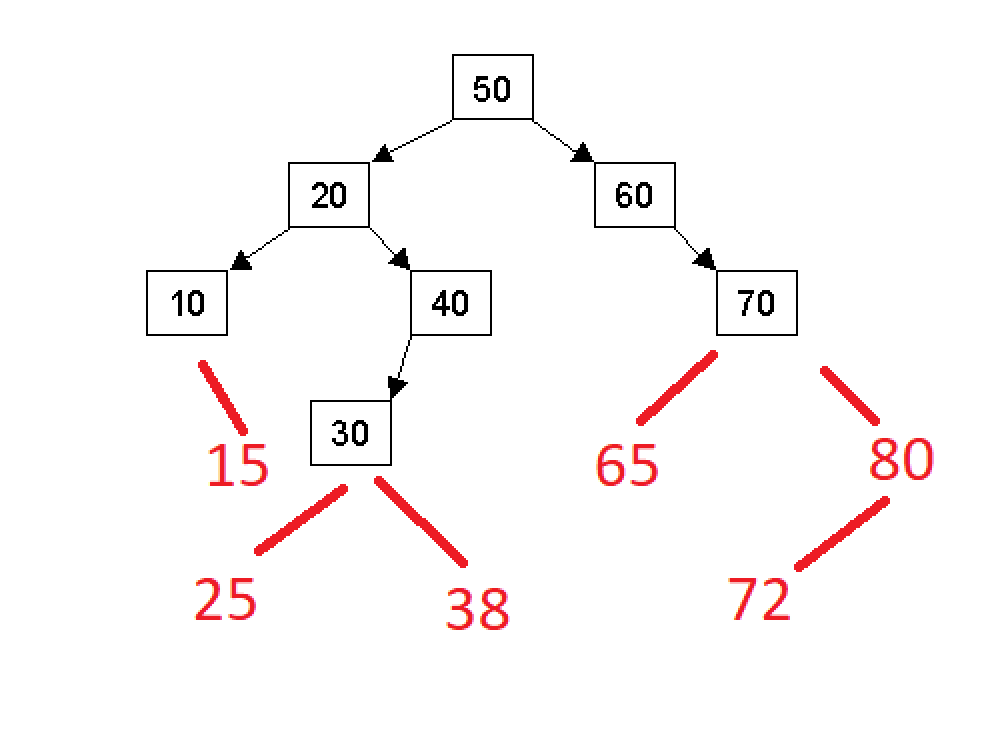
1a)

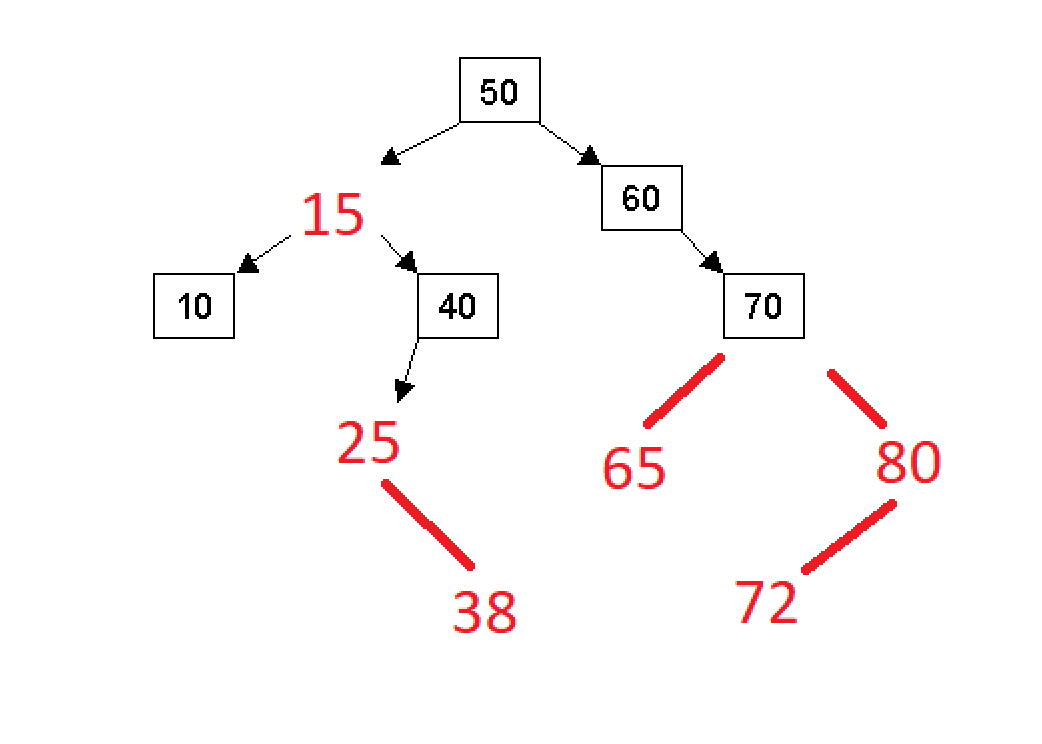
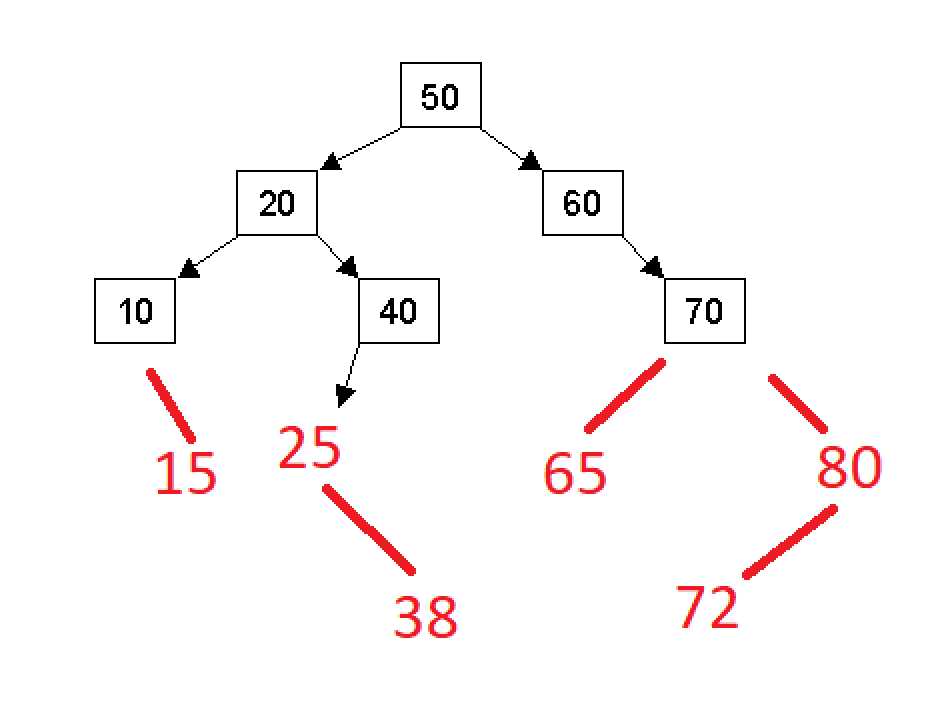
b)

Preorder: 50, 20, 10, 15, 40, 30, 25, 38, 60, 70, 65, 80, 72

Inorder: 10, 15, 20, 25, 30, 38, 40, 50, 60, 65, 70, 72, 80

Postorder: 15, 10, 25, 38, 30, 40, 20, 65, 72, 80, 70, 60, 50

c)



2a)

struct Node {

int val;

Node\* parent;

Node\* left;

Node\* right;

};

b)

If the tree is empty, create a new node with val and point root to it. Children and parent are null

Look at root node

Until insertion finished

If value to insert equals value of current node

do nothing (insertion finished)

If value less than value of current node

If there is a left child node

Look at left child node

Otherwise

add a new node, point current to it and point its parent to current. Children are null

If value is more than value of current node

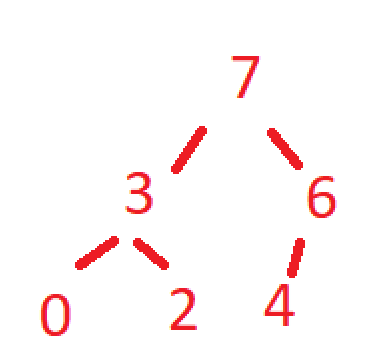
If there is a right child node

Look at right child node

Otherwise

Add a new node, point current to it and point its parent to current. Children are null

3a)



b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 3 | 6 | 0 | 2 | 4 |

c)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 3 | 4 | 0 | 2 |

4a) O(c + s)

b) O(log(c) + s)

c) O(log(c) + log(s))

d) O(log(s))

e) O(1)

f) O(log(c) + s)

g) O(s log(s))

h)O(c \* log(s))