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

post-order: 15 10 25 38 30 40 20 65 72 80 70 60 50

c. 

2.

1. struct Node {

public:

Node \* right;

Node \* left;

Node \* parent;

int value;

}

1. pseudocode: let n\_value be the value to insert

if tree is currently empty

initialize a new node and set its value to n\_value

set the tree’s root pointer to this node

return

traverse the tree starting from root

if current node’s value is less than n\_value

if current node has a left child, go left

else

allocate a new node and assign it n\_value

assign this new node as current node’s left child

set current node as the new node’s parent

return

else if current node’s value is greater than n\_value

if current node has a right child, go right

else

allocate a new node and assign it n\_value

assign this new node as current node’s right child

set current node as the new node’s parent

return

else

return //n\_value already exists in the tree



3.

2. 
3. 

4.

1. O(C + logS)
2. O(logC + S)
3. O(logC + logS)
4. O( logS)
5. O(1)
6. O(logC + S)
7. O(S)
8. O(ClogS)