1a.



1b.

Inorder: 10, 15, 20, 26, 30, 35, 40, 50, 60, 68, 70, 75, 80

Preorder: 50, 20, 10, 15, 40, 30, 26, 35, 60, 70, 68, 80, 75

Postorder: 15, 10, 26, 35, 30, 40, 20, 68, 75, 80, 70, 60, 50

1c.



2a.

| struct Node  {  int data;  \*Node left;  \*Node right;  \*Node parent;  } |
| --- |

2b.

| void insert(Node\* root, int value) {  If the root is a nullptr:  Create a new node with the given value and null pointers for the parent and children  Set the root node to be the newly created node  Else:  If the value is less than the value of the root node:  If the left child of the root node is a nullptr:  Create a new node with the given value, a pointer to the root node, and null pointers for the children  Set the left child of the root node to point to the newly created node  Else:  Recursive call - insert(left child of the root node, value)  If the value is greater than the value of the root node:  If the right child of the root node is a nullptr:  Create a new node with the given value, a pointer to the root node, and null pointers for the children  Set the right child of the root node to point to the newly created node  Else:  Recursive call - insert(right child of the root node, value)  Else:  Do nothing because the value is already in the tree |
| --- |

3a.



3b.

| 7 | 5 | 6 | 1 | 0 | 3 |
| --- | --- | --- | --- | --- | --- |

3c.

| 6 | 5 | 3 | 1 | 0 |
| --- | --- | --- | --- | --- |

4a. O(C + S)

4b. O(logC + S)

4c. O(logC + logS)

4d. O(logS)

4e. O(1)

4f. O(logC + S)

4g. O(SlogS)

4h. O(C \* logS)