1a.

50

20 60

10 40 70

15 30 65 80

25 35 75

1b.

50

25 60

10 40 70

15 35 65 80

75

1c.

In-order traversal: 10, 15, 20, 25, 30, 35, 40, 50, 60, 65, 70, 75, 80

Pre-order traversal: 50, 20, 10, 15, 40, 30, 25, 35, 60, 70, 65, 80, 75

Post-order traversal: 15, 10, 25, 35, 30, 40, 20, 65, 75, 80, 70, 60, 50

2a.

struct Node

{

int data;

Node\* parent;

Node\* leftChild;

Node\* rightChild;

};

2b.

insert(Node\* root, Node\* node)

If root is nullptr, set node’s parent pointer to nullptr and set node to root;

Else if node’s data is smaller than root’s data

If root doesn’t have a leftChild pointer

Set root’s leftChild pointer to node and set node’s parent pointer to root

Else

Call insert again with node and root’s leftChild pointer as root

Else if node’s data is greater than root’s data

If root doesn’t have a rightChild pointer

Set root’s rightChild pointer to node and set node’s parent pointer to root

Else

Call insert again with node and root’s rightChild pointer as root

3a.

7

3 6

0 1 4

3b. (front) 7, 3, 6, 0, 1, 4 (back)

3c. (front) 6, 3, 4, 0, 1 (back)

4.

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