50

20 60

10 40 70

15 30 65 80

25 36 74

b.

In-order: 10, 15, 20, 25, 30, 36, 40, 50, 60, 65, 70, 74, 80

Pre-order: 50, 20, 10, 15, 40, 30, 25, 36, 60, 70, 65, 80, 74

Post-order: 15, 10, 25, 36, 30, 40, 20, 65, 74, 80, 70, 60, 50

c.

50

15 60

10 40 70

25 65 80

36 74

2.

1. Struct node {

int data;

Node\* leftChild;

Node\* rightChild;

Node\* parent;

}

b. Pseudocode (passed in a V value)

if the tree is empty/root is null

Allocate new node and put V into it

Point root pointer to new node

Exit

Start at root pointer

While current pointer not null

If V is equal to current value

Exit

If V is less than current value

If current's left child isn't null

Pass in left child into function recursively

Else

Allocate new node and put V into it

Set current's left child to new node

If V is greater than current value

If current's right child isn't null

Pass in right child into function recursively

Else

Allocate new node and put V into it

Set current's right child to new node

3.



8

3 6

0 2 4

b. | 0 | 1 | 2 | 3 | 4 | 5 |

8 3 6 0 2 4

c.

6

3 4

0 2

4.

1. O(C + S)
2. O(log C + S)
3. O(log C + log S)
4. O(log S)
5. O(1)
6. O(log C + S)
7. O(S)
8. O(C \* log S)