1a. 50

20 60

10 40 70

15 30 65 80

25 38 72

1b. Preorder: 50 20 10 15 40 30 25 38 60 70 65 80 72

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

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

1c. 50

25 60

10 40 70

15 38 65 80

72

2a. Struct Node{

int val;

Node\* right;

Node\* left;

Node\* parent;

};

2b. if the tree is empty

* Create a new node and set the value
* Make root point to this node

otherwise

* If the value being added is equal to current node
  + Done
* If the value is less than the current node
  + If there is a left node
    - Move to left node
  + If not
    - Create a new node and set the value
    - Assign the current’s left to the new node
    - Assign the new node’s parent to the current node
* If the value is greater than the current node
  + If there is a right node
    - Move to right node
  + If not
    - Create a new node and set the value
    - Assign the current’s right to the new node
    - Assign the new node’s parent to the current node

3a. 7

3 6

0 2 4

3b.

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

3c.

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

4. a. O(C+S)

b. O(logC + S)

c. O(logC + logS)

d. O(logS)

e. O(1)

f. O(logC + S)

g. O(SlogS)

h. O(ClogS)