1. a) 50

20 60

10 40 70

15 30 64 80

23 36 75

b) Preorder: 50, 20, 10, 15, 40, 30, 23, 36, 60, 70, 64, 80, 75

Inorder: 10, 15, 20, 23, 30, 36, 40, 50, 60, 64, 70, 75, 80

Postorder: 15, 10, 23, 36, 30, 40, 20, 64, 75, 80, 70, 60, 50

c)

After Deleting 30:

50

20 60

10 40 70

15 36 64 80

23 75

After Deleting 30 & 20:

50

15 60

10 40 70

36 64 80

23 75

1. a) struct Node {

Node(int value) {

m\_value = value;

parent = nullptr;

right = nullptr;

left = nullptr;

}

int m\_value;

Node\* parent;

Node\* left;

Node\* right;

}

b) insert(int value) {

if tree is empty

allocate a new node with value

Have root node point to it

else

assign cur pointer to the root

if value equals cur node’s value, return

while both child nodes are not null

if value is less than the current node’s value

if the left child pointer is null

Set left child pointer to a new node with value

return;

else set cur node to cur node’s left child

if value is greater than the current node’s value

if the right child’s pointer is null

set right child pointer to a new node with value

return;

else set cur node to cur node’s right child

}

1. a)

7

5 6

3 1 2

b) [7, 5, 6, 3, 1, 2]

c) [6, 5, 2, 3, 1]

1. a) O(C + S)

b) O(log C + S)

c) O(log C + log S)

d) O(log S)

e) O(1)

f) O(log C + S)

g) O(SlogS)

h) O(log S \* C)