Komal Prakash

HW 5

304980756

1. A)

50

20

60

10

40

70

30

65

80

15

76

34

25

B) In-order: 10, 15, 20, 25, 30, 34, 40, 50, 60, 65, 70, 76, 80

Pre-order: 50, 20, 10, 15, 40, 30, 25, 34, 60, 70, 65, 80, 76

Post-order: 15, 10, 25, 34, 30, 40, 20, 65, 76, 80, 70, 60, 50

C)

50

15

60

10

40

70

25

65

80

76

34

1. a) struct Node {

Node(int val){

data=val;

left=right=NULL;

}

int data;

Node\* left,right,parent;

}

b) void insert(Node\* root, Node\* newNode)

if root is nullptr

set root to newNode

set newNode’s parent to nullptr

else if newNode’s data>root’s data

if root’s right child =NULL

set root’s right child to newNode

set newNode’s parent to root

else

call insert(root’s right child, newNode)

else if newNode’s data<root’s data

if root’s left child =NULL

set root’s left child to newNode

set newNode’s parent to root

else

call insert(root’s left child, newNode)

1. a)

7

3

5

4

2

0

3b) array = {7, 3, 5, 0, 2, 4}

3c)

5

3

4

2

0

4)

a) O(C+S)

b) O(logC + S)

c) O(logC + logS)

d) O(1+logS) = O(logS)

e) O(1+1) = O(1)

f) O(logC + S)

g) O(1+SlogS+S) = O(SlogS)

h) O(ClogS)

5b) Without the second string variable parameter, you would not have a way to keep track of the previous class names which is necessary in order to approach this function recursively.