CS32 HW5

1a. 50

20 60

10 40 70

15 30 62 80

75

1b:

Inorder traverse print out: 10,15, 20,30,40, 50, 60, 62, 70, 75, 80

Preorder traverse print out: 50, 20, 10, 15, 40, 30, 60, 70, 62, 80, 75

Postorder traverse print out: 15, 10, 30, 40, 20, 62, 75, 80, 70, 60, 50

1c: 50

40 60

10 70

15 62 80

75

2.

a:

struct Node

{ int m\_num;

Node\* left;

Node\* right;

Node\* parent;

}

b:

First, implement a helper function called insertHelp that takes in two parameters. First one is the root pointer and the second is the value of which we want to insert.

Void insertHelp(Node\* root, const int& k){

If root is nullptr(tree is empty) 🡪 create a root node with both children pointer as null and value as k.

If root’s value is > k 🡪 insertHelp(root -> left);

If root’s value is < k 🡪 insertHlep(root -> right);

Otherwise, do nothing since it’s already that value is already Inside the tree.}

3.

a. 7

5 6

4 0 3

b. array representation: 7 5 6 4 0 3

c. 6 5 3 4 0

4.

a. O(C+logS)

b. O(C+S)

c. O(logC+S)

d. O(1+logS)

e. O(1): O(1+1) which is O(2) an O(constant)

f. O(logC + S)

g. O(SlogS)

h. O(ClogS): //loop through every class O(C), and for each class, binary search that student’s name O(logS).