Adam Cole

Homework 5

1.a. 50

20 60

10 40 70

15 30 65 80

25 37 73

b. In-order: 10, 15, 20, 25, 30, 37, 40, 50, 60, 65, 70, 73, 80

Pre-order: 50, 20, 10, 15, 40, 30, 25, 37, 60, 70, 65, 80, 73

Post-order: 15, 10, 25, 37, 30, 40, 20, 65, 73, 80, 70, 60, 50

c. 50

15 60

10 40 70

25 65 80

37 73

2.a. struct TreeNode {

int value;

TreeNode\* left;

TreeNode\* right;

TreeNode\* parent;

};

b. void insert( int num ) {

if the root is the null pointer {

There are no elements in the BST.

Set the root equal to a new node and its left, right, and parent TreeNode\*

pointers as the null pointer. Then return.

}

Set a temporary TreeNode\* equal to the root.

Run until return is called {

if num is equal to the temp’s value {

return, the item is already in the tree.

} else if num is less than the temp’s value {

if the temp’s left subtree is not empty {

shift the temp TreeNode\* pointer to the left subtree’s root

} else {

Set the left subtree pointer equal to a new node and its left,

And right, TreeNode\* pointers as the null pointer.

Set the parent TreeNode\* pointer equal to the temp pointer.

Then return.

}

} else the num is greater than the temp’s value {

if the temp’s right subtree is not empty {

shift the temp TreeNode\* pointer to the right subtree’s root

} else {

Set the right subtree pointer equal to a new node and its left,

And right, TreeNode\* pointers as the null pointer.

Set the parent TreeNode\* pointer equal to the temp pointer.

Then return.

}

}

}

}

3.a. 8

3 6

0 2 4

b. { 8 , 3 , 6 , 0 , 2 , 4 }

c. 6

3 4

0 2

4.a. O ( C + S )

b. O ( S + log2C )

c. O ( log2C + log2S )

d. O ( log2S )

e. O ( 1 )

f. O ( S + log2C )

g. O ( S \* logS )

h. O ( C \* log2S )