Zane Koch

CS32

HW5

Question 1:

A: before any insertions

50

20 60

10 40 70

30

-after inserting 80

50

20 60

10 40 70

30 80

-after inserting 80 and 65

50

20 60

10 40 70

30 65 80

-after inserting 80, 65 and 73

50

20 60

10 40 70

30 65 80

73

-after inserting 80, 65, 73, and 15

50

20 60

10 40 70

15 30 65 80

73

-after inserting 80, 65, 73, 15, and 37

50

20 60

10 40 70

15 30 65 80

37 73

-after inserting 80, 65, 73, 15, 37, and 25

50

20 60

10 40 70

15 30 65 80

25 37 73

B:

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

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

Inorder: 10, 15, 20, 25, 30, 37, 40, 50, 60, 65, 70, 73, 80

C:

50

25 60

10 40 70

15 30 65 80

37 73

Problem 2:

A:

Struct node

{

Node\* left;

Node\* right;

Node\* parent;

Int data;

}

B:

Node \* binaryTree::insert(node\* node, int value) //start node and value

{

If (node == nullptr)

return node(value); //returns a new node with no pointers because this is the first node in the tree

if ( value < node->value)

{

Node \* leftChild = insert ( node->left, value);

Node -> left = leftChild;

leftChild->parent = node;

}

Else if ( value > node->value)

{

Node \* rightChild = insert ( node->right, value);

Node -> right = rightChild;

rightChild->parent = node;

}

Return node;

}

Problem 3:

A:

8

3 6

0 2 4

B:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| index | 0 | 1 | 2 | 3 | 4 | 5 |
| value | 8 | 3 | 6 | 0 | 2 | 4 |

C:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 |
| value | 6 | 3 | 4 | 0 | 2 |

Problem 4:

A:

O( C + S)

B:

O( logC + S)

C:

O(logC + logS)

D: ?

O( 1 + logS ) same as O(logS)

E:

O( 1 )

F:

O(logC + S)

G: ?

O( 1 + SlogS) same as O(SlogS)

H:

O(ClogS)