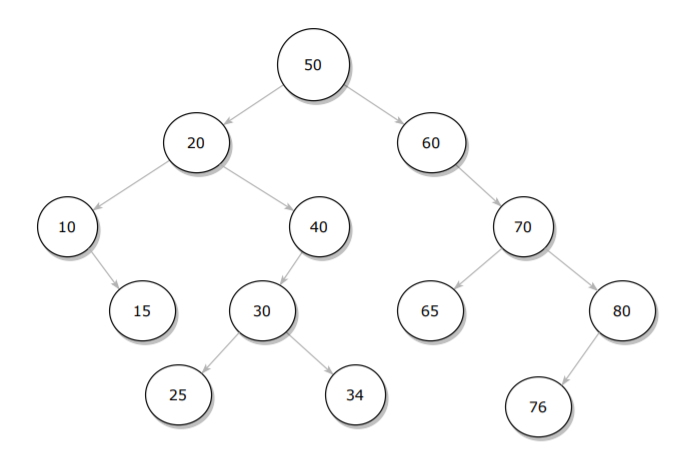
Jaden Booher

405-333-407

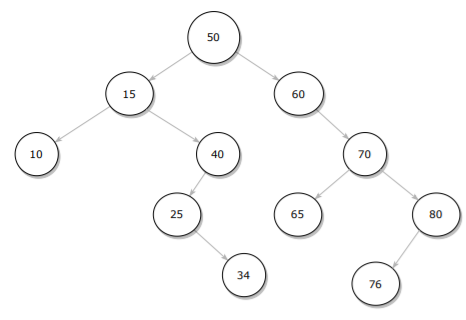
Homework 5

­­1a.

1b. **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

1c.

2a.

struct node

{

int value;

node \*leftChild, \*rightChild, \*parent;

};

2b.

void insert(int value)

{

if (root node is equal to nullptr)

root = new node with passed value

Return

Create a pointer node and point it to root

While (true)

If current node’s value == value (Node already exists)

Return

If value is less the current node’s value

If current node has no left child

Create new node with passed value

Return

Else

Set current node to current node’s left child

Return

Else if value is greater than current node’s value

If current node has no right child

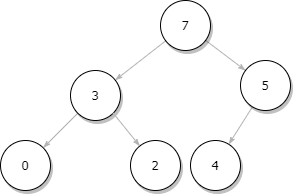
Create new node with passed value

Return

Else

Set current node to current node’s right child

Return

};

3a.

3b.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 7 | 3 | 5 | 0 | 2 | 4 |

3c.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 5 | 3 | 4 | 0 | 2 |

4a. O(C+logS)

4b. O(S + logC)

4c. O(logC + logS)

4d. O(logS)

4e. O(1)

4f. O(logC + S)

4g. O(SlogS)

4h. O(ClogS)

5b. You would not be able to solve this problem with a one-parameter “listAll” function because you would not be able to recursively call the function and make changes to each string output at the same time. The string needed to be safe upon each call of “listAll” during recursion.