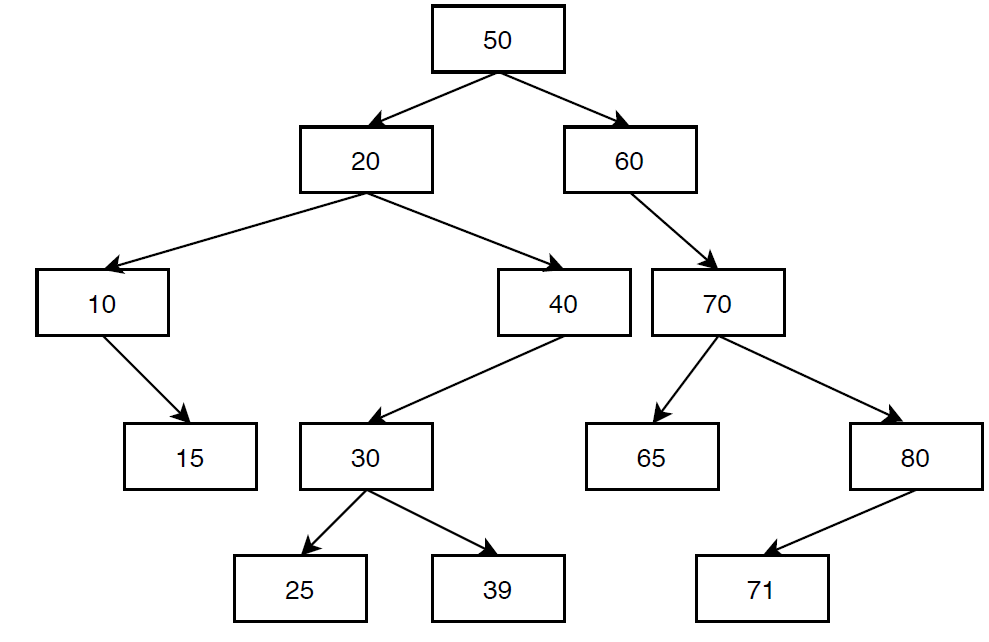
Danning Yu

6/3/19

Homework 5

1(a) See picture below.



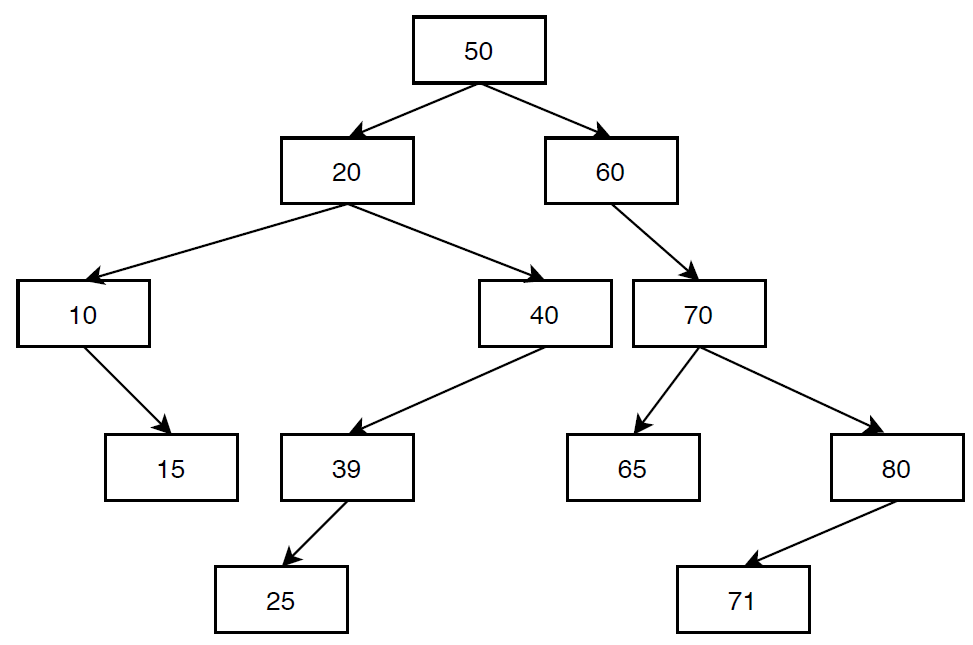
1(b) In-order: 10 15 20 25 30 39 40 50 60 65 70 71 80

Pre-order: 50 20 10 15 40 30 25 39 60 70 65 80 71

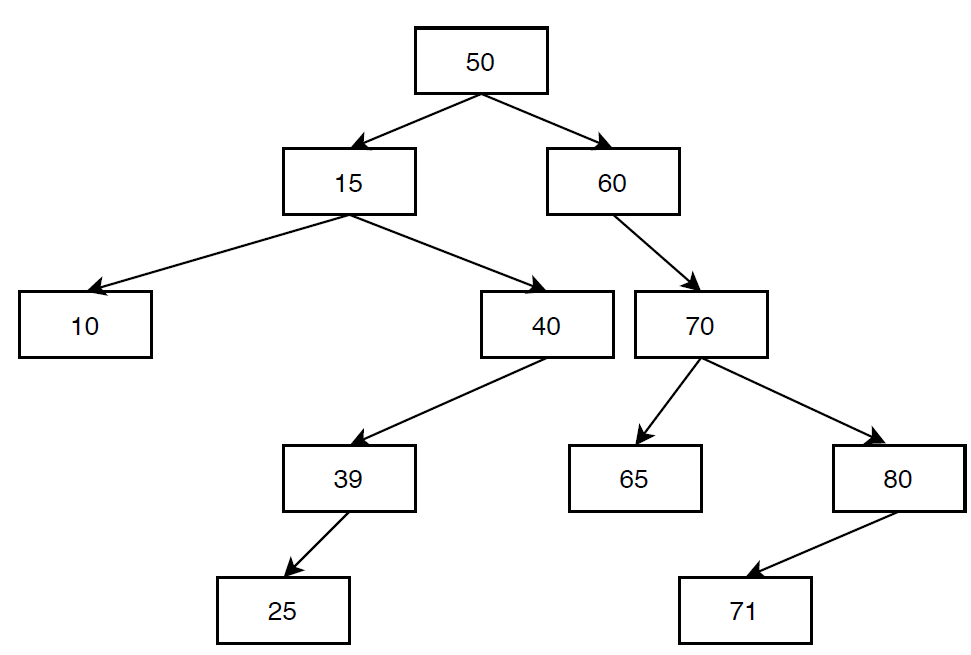
Post-order: 15 10 25 39 30 40 20 65 71 80 70 60 50

1(c) See pictures below.

After removing 30:



After removing 20:



2(a)

struct Node{

Node\* m\_parent;

Node\* m\_right;

Node\* m\_left;

int m\_data;

Node(int data, Node\* parent):m\_parent(parent), m\_right(nullptr), m\_left(nullptr), m\_data(data){}

};

2(b) This is for the case where toInsert is a pointer to a Node that already contains the value to be inserted. The function is written this way as the answer because the spec asks to insert a “new node,” not a new value, and toInsert is the node that is to be inserted.

void insertNode(Node\* root, Node\* toInsert){

if root is nullptr:

//empty BST

set root to toInsert

set toInsert’s m\_parent to nullptr

return;

if root is not nullptr and both root’s left and right nodes are nullptrs

//reached bottom of tree

if toInsert’s data is >= root’s data:

root’s right pointer points to ToInsert

toInsert’s parent node points to root

return;

else if toInsert’s data is < root’s data:

root’s left pointer points to ToInsert

toInsert’s parent node points to root

return;

if toInsert’s data >= root’s data

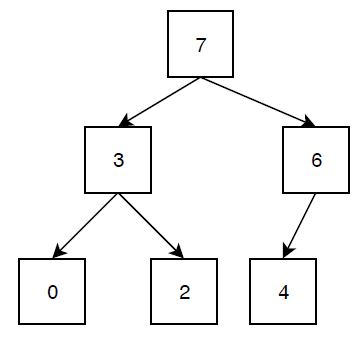
call insertNode(root->m\_right, toInsert)

else if toInsert’s data < root’s data

call insertNode(root->m\_left, toInsert)

}

3(a) See diagram below.



3(b)

First row is index, second row is values.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 |
| Values | 7 | 3 | 6 | 0 | 2 | 4 |

3(c)

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

4(a) O(S+C)

4(b) O(logC+S)

4(c) O(logC+logS)

4(d) O(1+logS), which simplifies to O(logS)

4(e) O(1+1), which simplifies to O(1)

4(f) O(logC + S)

4(g) O(1+SlogS), which simplifies to O(SlogS)

4(h) O(logS + C)