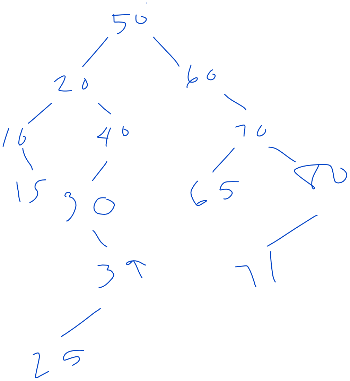
Name: Morgan Lim

UID: 105 168 668

1

a)



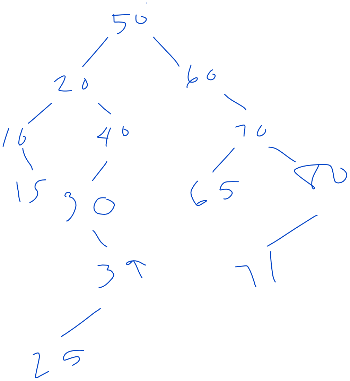


b) in order: 10, 15, 20, 25, 30, 39, 40, 50, 60, 65, 70, 71, 80

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

c)





2

a)

class Node

{

public:

Node(int data)

: m\_data(data), m\_parent(nullptr), m\_lChild(nullptr), m\_rChild(nullptr){}

Node(int data, Node\* parent, Node\* lChild, Node\* rChild)

: m\_data(data), m\_parent(parent), m\_lChild(lChild), m\_rChild(rChild) {}

void setParent(Node\* p) {m\_parent = p;}

Node\* getParent(Node\* p) {return m\_parent;}

void setRChild(Node\* p) {m\_rChild = p;}

Node\* getRChild(Node\* p) {return m\_rChild;}

void setLChild(Node\* p) {m\_lChild = p;}

Node\* getLChild(Node\* p) {return m\_lChild;}

private:

int m\_data;

Node\* m\_parent, m\_lChild, m\_rChild;  
};

b)

if the subtree is empty

return the new node pointer

otherwise if the item in the subtree is greater than the new node’s value

insert the new pointer to the left

set parent of new node to subtree pointer

otherwise

insert the new pointer to the right

set parent of new node to subtree pointer

3

a)



b) 7 3 6 0 2 4

c) 6 3 4 0 2

4

a) O(C+S)

b) O(log C + S)

c) O(log C + log S)

d) O(log S)

e) O(1)

f) O((log C) + S)

g) O(S log S)

h) O(C log S)