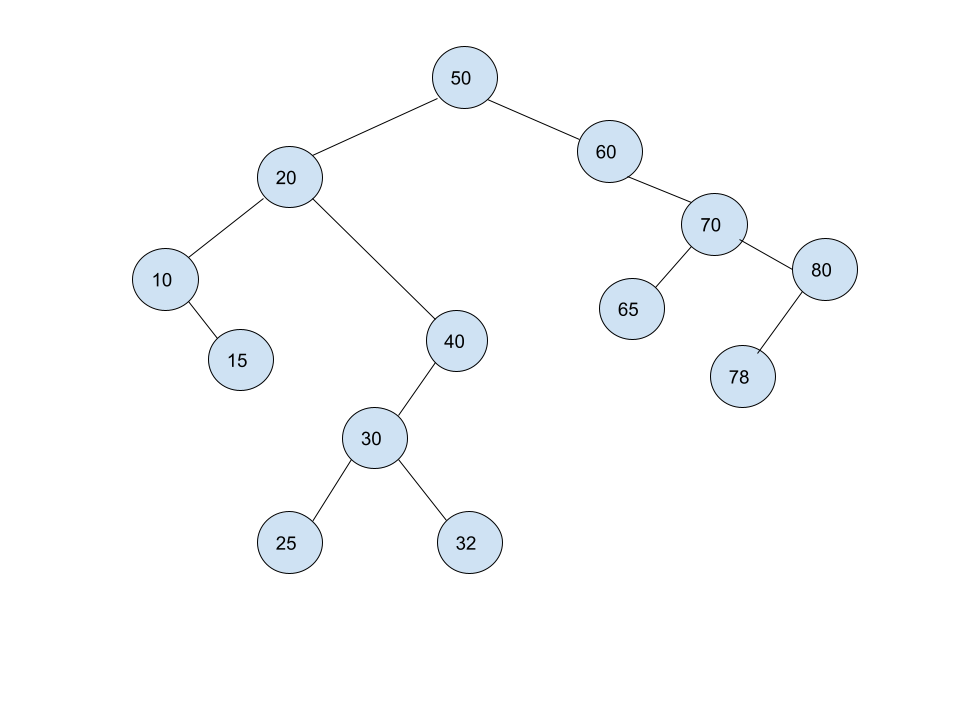
Darren Wu CS32 HW5

1a)



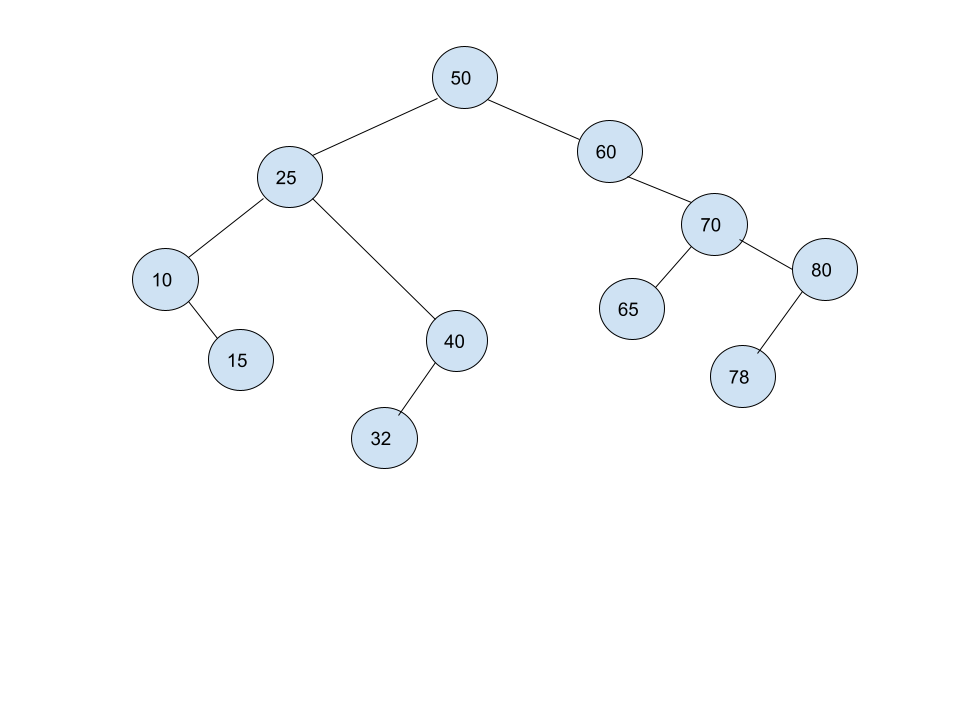
1b)

Pre: 50,20,10,15,40,30,25,32,60,70,65,80,78

In: 10,15,20,25,30,32,40,50,60,65,70,78,80

Post: 15,10,25,32,30,40,20,65,78,80,70,60,50

1c)



2a)

struct Node{

Node\* parent;

Node\* leftChild;

Node\* rightChild;

int value;

};

2b)

If tree is empty

Set the root to a new node with a value, all pointers to nullptr

Return

Initialize a temporary node to the root

While you still need to insert(loop)

If value is equal to temp’s value

return

Else If the value is less than the temp node’s value

If temps left child isn’t null

Set temp to the left child

Else

Create a new node of value with left, right children, nullptr,

Set the created node to temp’s left child

Set the new node’s parent to temp

return

Else if value is greater than temp node’s value

If temp’s right child isn’t null

Set temp to the right child

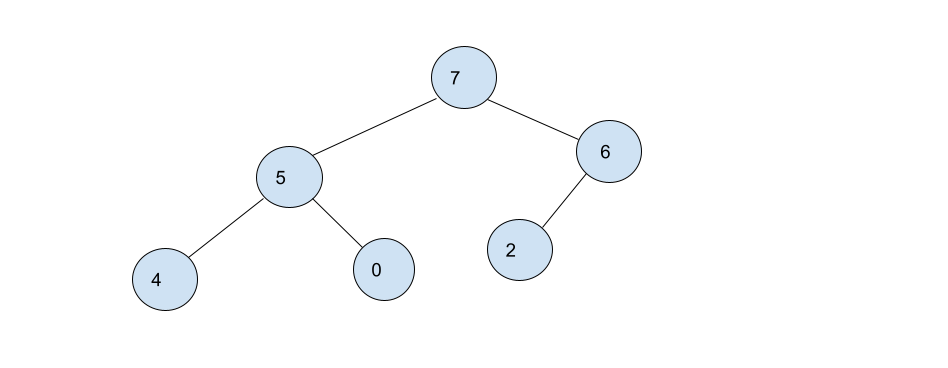
Else

Set a created node of value to temp’s right child

Set the created node’s children to null, set its parent pointer to temp

Return

3a)

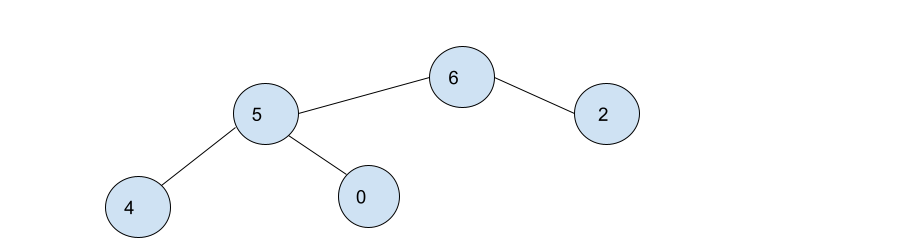


3b)

Array value [7,5,6,4,0,2]

Position 0 1 2 3 4 5

3c)



Array value [6,5,2,4,0]

Position 0 1 2 3 4

4)

a) O(C+S)

b) O(logC + S)

c) O(logC+logS)

d) O(1+logS) = O(logS)

e) O(1+1) = O(1)

f) O(logC + S)

g) O(1+SLogS) = O(SLogS)

h) O(C\*LogS)