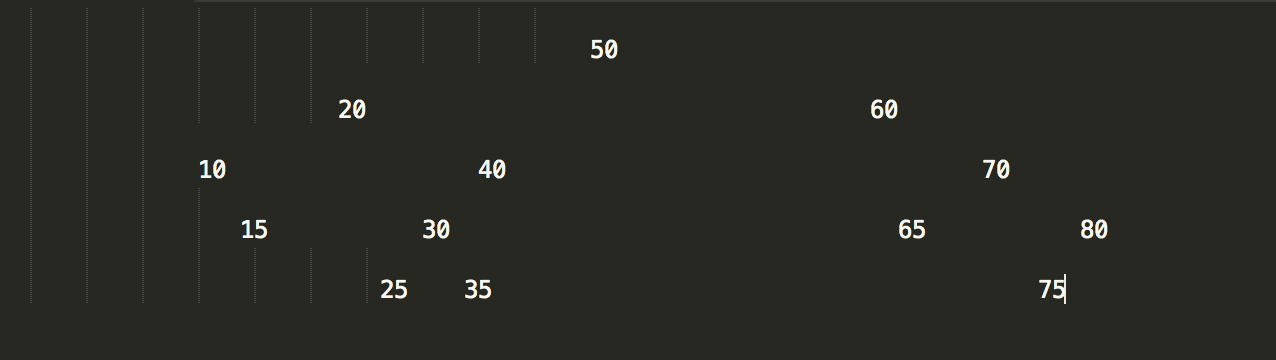
1a) I had a lot of trouble creating the tree on Microsoft Word, so I created it on sublime text. Here is a screenshot, as I had more trouble transferring it over with the correct formatting.



1b)

Pre-Order Post-Order

50 15

20 10

10 25

15 35

40 30

30 40

25 20

35 65

60 75

70 80

65 70

80 60

75 50

In-Order

10

15

20

25

30

35

40

50

60

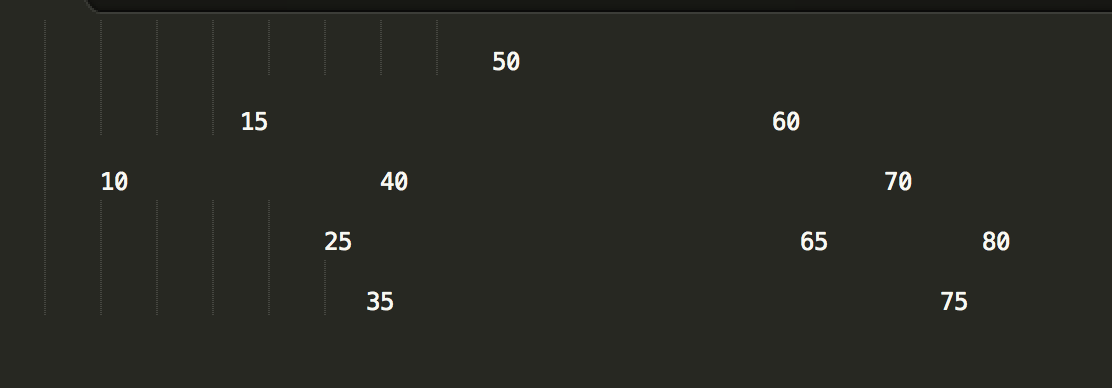
65

70

75

80

1c) Deleting the 30 and 20 nodes in that respective order would produce this tree.



2a)

struct Node

{

Node(int val, Node\* par)

{

value = val;

left = right = nullptr;

parent = par;

}

Node\* left;

Node\* right;

Node\* parent;

int value;

};

2b)

void Insert (int value)

{

InsertAlgo(root, value, nullptr);

}

void InsertAlgo(Node\* cur, int value, Node\* parent)

{

if (cur equals nullptr)

create a new node, store the value in it, make the child pointers null, and make its parent the parent in the parameter

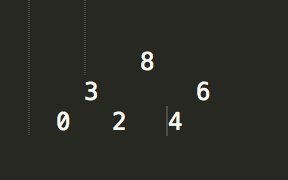
else if (value is less than cur’s value)

Insert(cur->left, value, cur)

else if (value is greater than cur’s value)

Insert(cur->right, value, cur)

}

3a) The heap would look as follows:

3b) The array form of this heap would be {8, 3, 6, 0, 2, 4}

3c) The array that results from executing remove again is {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)

h. O(C\*log(S))