==========

CS311 Yoshii - HW5 Binary Heap and Balanced Trees (based on week 9 and 10)

==========

DUE: Week 11 Monday

TOTAL 30 points **Your score is:**

Your Name:Edaurdo Martinez

Date turned in:11/6/16

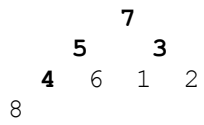
Purpose: To be able to follow the algorithms for heap and balanced trees.

No Programming is required!

Heap Tree [12 pts] **Your score is:**

Problem 1) [2 per drawing = 8 pts]

Create a new Heap Tree using Heapify given the following:



We heapify each sub-tree starting from the bottom sub-trees.
(i.e., the one at 4, the one at 3, then the one at 5 then the one at 7)
For each heapify, we make the root trickle down
to the right place by continuously exchanging it with
the smaller of the children, until no child is smaller.

Intermediate results: Draw the tree after each sub-tree is heapified.
And finally show the heap tree. (4 drawings)

Problem 2) [4 pts]
Then, remove 1 from the root and do re-heapify => show the resulting tree.

See EC1 Printer Queue for implementing heap using an array. Highly Recommended.

AVL [9 pts] **Your score is:**

Given ((4 5 (6 7 8)) 9 10) where 4,5*,6,7*,8,9*,10 are the nodes. I am putting * next to the nodes which you can think of as operators.

Draw the corresponding tree and indicate the balance factor for each internal node.
(i.e. 0, +1, -1, +2 or -2) [3pts]

Identify its case (case 1,2,3,or 4 of the Notes) [2pts]

What is the balanced expression after the required rotation is done? [2pts]

And then draw the rotated/balanced tree. [2pts]

B-tree [9 pts] **Your score is:**

Start with this 2-3 tree (i.e. M=3).

```
            GK

    C       I       M

 AB    DE   H  J    L    NO
000   000   00 00   00    000
```

Add P. Show the resulting tree. [3pts]

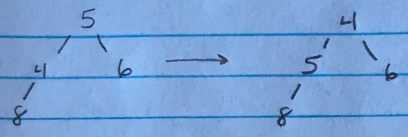
Then, add Q to the result. Show the resulting tree. [3pts]

Finally, add R. Show the resulting tree which has P and Q and R. [3pts]

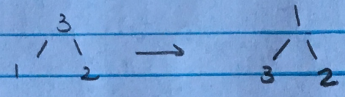
Submit this file:
Drawings can be done by hand and scanned or photographed.
- This assignment sheet with your answers and pictures inserted into the file.

problem 1: 1. stays as is

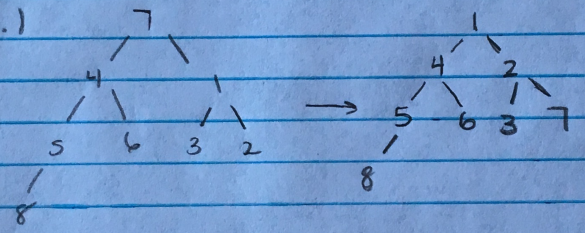
2.)



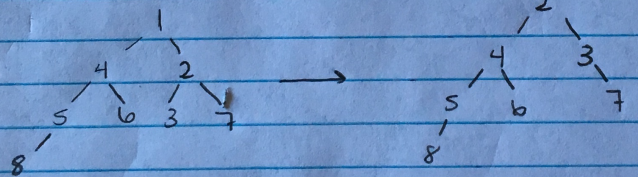
3.)



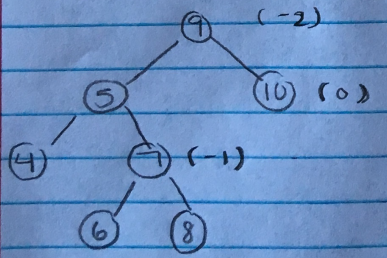
4.)



problem 2:

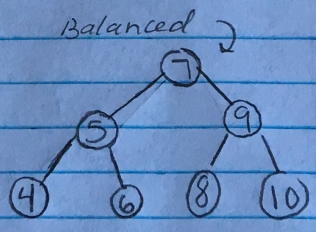


AVL:



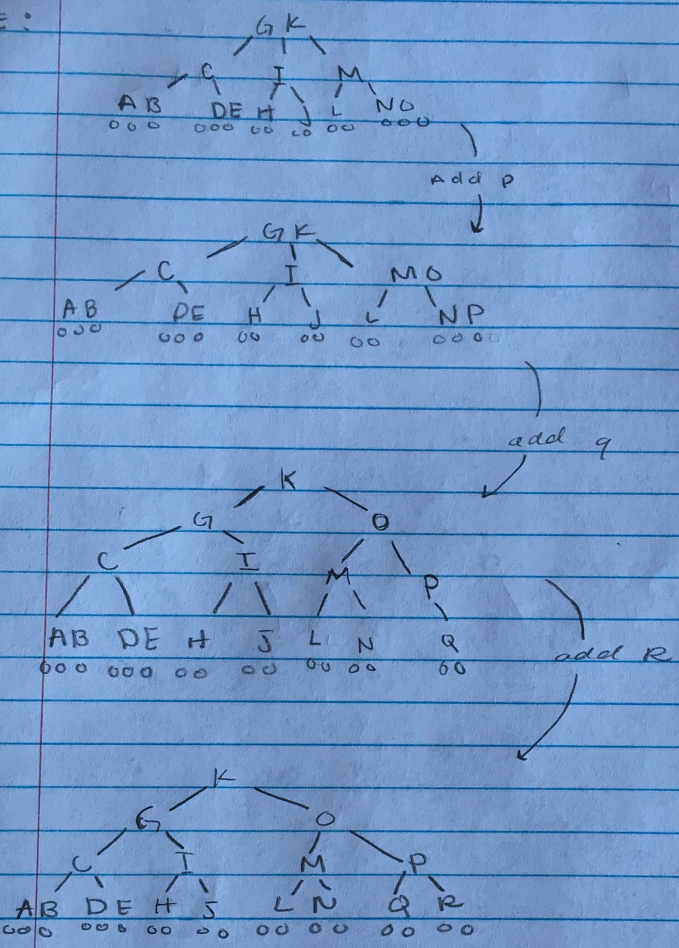
case 4

Balanced



((4 5 6) 7 (8 9 10))

B-TREE:


G K
C I M
A B D E H J L N O
add p
G K
C I M O
A B D E H J L N P
add q
K
G O
C I M P
A B D E H J L N Q
add R
K
G O
C I M P
A B D E H J L N Q R