2018-10-09 Midterm & HWs Review

Tuesday, October 9, 2018 8:57 AM

What's on the exam?

- Algorithm analysis questions
  - E.g. what is the efficiency of this loop?

```
For(int i = 0; i < input.length(); i++){cout << "Hello"; << endl; }
```

  - For the above, list the number of times "Hello" will be printed when input is "a", "aa", "aaa", etc.
  - Also, "natural" language efficiency questions
    - E.g. A runner is running laps around a track. At the start marker of the track is a stack of pennies. He must continue to run around the track until he has collected all of the pennies. Each lab, the runner is allowed to pick up twice as many pennies as he collected on the prior lap. How many laps must the runner run?
- Conceptual data structures questions from HWs
  - Drawing trees from traversals
  - State of stack / queue after X operations
  - Conceptual tree questions (e.g. what is an internal node)
  - Draw AVL tree after insert / removal
  - Binary heaps
  - Binomial heaps
  - Skew heaps

	1	2	3	4	5	6	7	8	9	10	11
LogN	0	1	1.5	2	2.3	2.6	2.8	3	3.1	3.3	3.5
N	1	2	3	4	5	6	7	8	9	10	11
NLogN	1	2	5	8	12	13	14				
N^2	1	2	9	16	25	36	49	64			

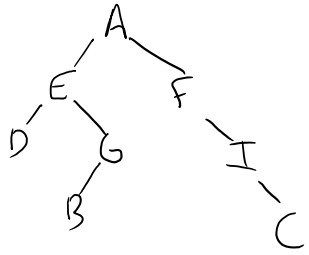
CS 211 Homework #1

Please complete the homework problems. Note that this is an individual assignment and all work must be your own. Be sure to show your work when appropriate. This assignment is due **in class** on Thursday, September 27, 2018.

1. [3] Given the following pre-order and in-order traversals, reconstruct the appropriate binary tree. **NOTE: You must draw a single tree that works for both traversals.**

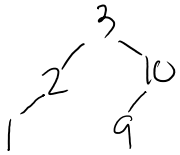
Pre-order: A, E, D, G, B, F, I, C

In-order: D, E, B, G, A, F, I, C



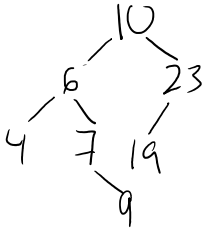
2. [3] Starting with an empty BST, show the result of the following sequence of operations. Assume that all removals come from the left subtree when the node to remove is full.

Insert(5), Insert(10), Insert(2), Insert(9), Insert(1), Insert(3), Remove(5).



3. [3] Starting with an empty BST, show the result of the following sequence of operations. Assume that all removals come from the right subtree when the node to remove is full.

Insert(10), Insert(5), Insert(23), Insert(4), Insert(19), Insert(7), Insert(9), Insert(6), Remove(5).



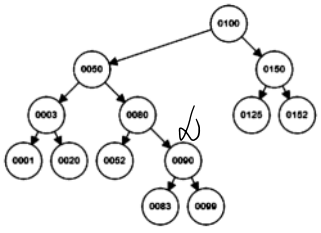
4. [2] Assume that we have two algorithms that accomplish the same task. Algorithm A has a non-simplified runtime of O(n^2 + 5n) and Algorithm B has a runtime of O(n^3). Which should we use? Why?

5. [3] Provided are counts of computation for various series. Indicate the most likely runtime complexity for each:

	N = 1	N = 2	N = 3	N = 4	N = 5	N = 6	N = 7	N = 8
Alg. A	6 8	14 10	24 12	36 14	50 16	66 18	85 ~~14~~	104
Alg. B	2 6	8 10	18 14	32 18	50 20	70 29	99 ~~36~~	128
Alg. C	1 3	4 5	9 7	16 7	23 8	31 8	39 9	48
Alg. D	1 6	7 4	11 5	14 3	17 2	19 2	21 1	22
Alg. E	2 2	4 2	6 2	8 2	10 2	12 2	14 2	16

n^2
n^2
$N \log N$
$\log N$
N

6. Given the following binary tree:



A. [1] What is the height of the tree?

4

B. [1] What is the depth of node 90?

3

C. [1] What is the height of node 90?

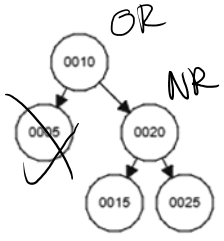
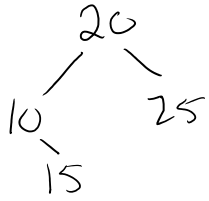
1

D. [3] Give the pre-order, in-order, and post-order traversal of this tree.

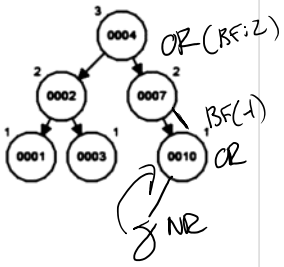
CS 211 Homework #2

Note that this is an individual assignment and all work must be your own. Be sure to show your work when appropriate. This assignment is due **in class** on Tuesday, October 9, 2018.

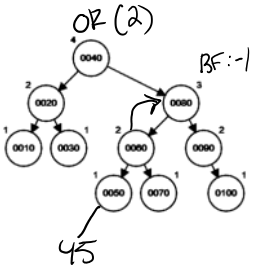
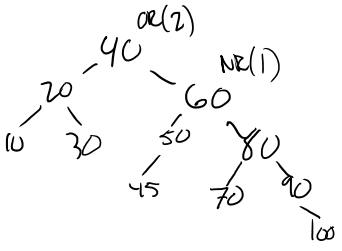
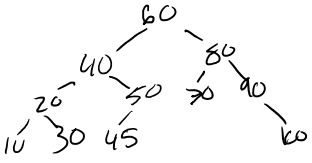
1. [3] Remove 5 from the following AVL tree; draw the results:

Original	**After removal**
	
After 1st Rotation	**After 2st Rotation (may or may not be required)**

2. [3] Insert the value "8" into the following AVL tree. Draw the result:

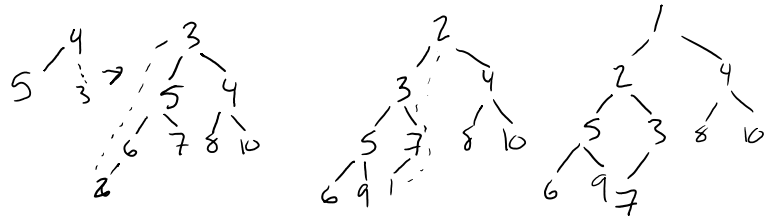
Original	After removal
	
After 1st Rotation	**After 2st Rotation (may or may not be required)**
	

3. [3] Add 45 to the following AVL tree.

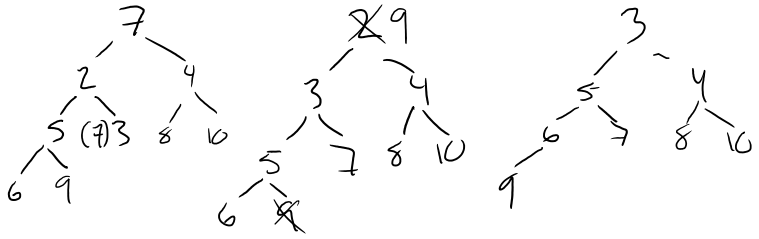
Original	After insert
	
After 1st Rotation	**After 2st Rotation (may or may not be required)**
	

4. **Binary Heaps** Starting with an empty binary **min heap**, show the following.

A. [3] The final state of the heap, <u>in tree form</u>, after adding in the values: 5, 4, 3, 6, 7, 8, 10, 2, 9, 1



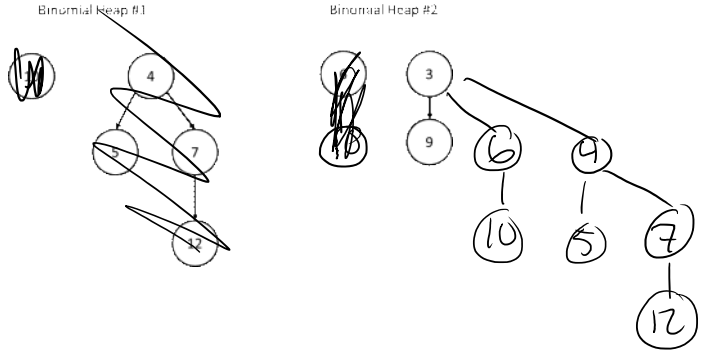
B. [2] The state of the heap, <u>in tree form</u>, after two Dequeue() operations



C. [1] The final, array-based version of the heap

3	5	4	6	7	8	10	9		
0	1	2	3	4	5	6	7	8	9

5. [3] Merge the following two **<u>binomial heaps</u>**



6. [3] Enqueue 10 to the following **<u>skew heap</u>**. Draw the results.