EC 504 – Sample Midterm

Put your name and BU ID on the first page. This exam is closed book and no notes – except for 2 pages of personal notes to be passed in at the end. No use of laptops or internet.

COMMENT: This sample is the style but of course details may change and some maybe almost the same! Don't have to do everything to get a good score.

1. (20 pts) Answer True or False to each of the questions below. Each question is worth 2 points. Answer true only if it is true as stated, with no additional assumptions. No explanation is needed, but any explanation is likely to earn partial credit, and no explanation will not earn any credit if the answer is wrong.

   (a) The best median finding algorithm is on average $\Theta(N)$ but worst case $\Theta(N^2)$

   (b) The sum total height $T_H(N)$ and total depth $T_D(N)$ of N nodes in a binary tree is HN only if the tree is perfect.

   (c) All heaps can be represented by full trees.

   (d) For a binary tree of (total) height H the number $(L(H))$ of leaves obeys $L(H) = O(2^H)$.

   (e) It is possible to formulate Quick Sort to be worst case $O(N \log N)$ including the cost of picking a suitable pivot.

   (f) A connected undirected acyclic graph is a binary tree.

   (g) Given an array of N integers, the best algorithm has a worst case time to build the heap $\Theta(Nlog(N))$.

   (h) The minium of $\sum_{i=1}^{N} i * a[i-1]$ for all permutation of elements in $a[i]$ is sort $a[i]$ in descending order (i.e. larger one first).

   (i) The two traversal BFS (bread first search) and DFS (depth first search) for a graph $G(A, N)$ can implemented by entering the nodes one at time into Stack and Queue data structure respectively.

   (j) $\sum_{i=1}^{N} i^k \in O(N^k)$.

2. (10 pts) Consider the following tree:

```
       A
     /  \
    B     C
   /     / \
  D     E   F
 / \         \
G  H          I
               \
                J
```

(a) List the height and depth of each of the nodes A-J

(b) List the nodes in pre-order, in-order and post-order.

3. (20 pts) Consider the following min-heap.

```
                 3
             /           \
           8               4
        /    \          /     \
      10       9       7       6
     /  \    /  \   /  \   / \
   15    11 2019 13 15  20 10 14
  / \   /
17 16 12
```

(a) Show the min-heap which results after inserting the element 2 in the heap. (Indicate the sequence of steps with arrows.) Then in this new heap show the steps required to delete element 3. Draw the final min-heap

(b) Consider min-heap as an array **int a[N+1]** with the size of heap stored in **a[0] = N**. (For example the one above has **a[0] = 18** and $a[1], a[2], \cdots, a[18]$. Describe carefully an $\Theta(N \log N)$ algorithm to sort `in place` the array elements $a[1], a[2], \cdots, a[N]$ in descending order. Use $a[0]$ to store the decreasing number remaining in the heap as you proceed with the sort.

(c) Now use $a[0]$ as a temporary to re-arrange `in place` the sort into ascending order with $O(N)$ extra swaps. Be brief but explicit on the method.

(d) Re-arrange the original min-heap (repeated below) into a max-heap by a "bottom up" $O(N)$ algorithm. Describe the steps level by level.

```
                 3
             /           \
           8               4
        /    \          /     \
      10       9       7       6
     /  \    /  \   /  \   / \
   15    11 2010 13 15  20 10 14
  / \   /
17 16 12
```

4. (20 pts) This problem is to construct step by step BST and AVL trees given the following list of $N = 10$ elements.

$$28 \quad 14 \quad 7 \quad 4 \quad 6 \quad 30 \quad 36 \quad 33 \quad 10 \quad 40$$

(a) Insert them sequentially into a BST (Binary Search Tree).

(b) Insert them sequentially into an empty AVL tree, restoring the AVL property after each insertion. Show the AVL tree which results after each insertion and name the type of rotation (RR or LL zig-zig or versus RL or LR zig-zag).

(c) Relative to the BST has AVL tree decreased the total height $T_H(N)$ and the total depth $T_D(N)$? Give the amont that these have change.

(d) For $N = 2^{16} - 1$ what is the max and minimum values for $T_H(N) + T_D(N)$.

5. (15 pts) You are interested in compression. Given a file with characters, you want to find the binary code which satisfies the prefix property (no conflicts) and which minimizes the number of bits required. As an example, consider an alphabet with 8 symbols, with relative weights (frequency) of appearance in an average text file give below:

alphabet:	G	R	E	A	T	F	U	L
weights:	8	6	60	30	4	12	16	15

(a) Determine the Huffman code by constructing a tree with **minimum external path length**: $\sum_{i=1}^{8} w_i d_i$. (Arrange tree with smaller weights to the left.)

(b) Identity the code for each letter and list the number of bits for each letter and compute the average number of bits per symbole in this code. Is it less than 3? (You can leave the answer as a fraction since getting the decimal value is difficult without a calculator.)

(c) Give an example of weights for these 8 symbols that would saturate 3 bits per letter. What would the Huffman tree look like? Is a Huffman code tree always a full tree?

6. (15 pts) For each of these recursions, please give the tightest upper bound for the recursion. You can write your answer as $T(n) \in O(g(n))$ for your best choice of function g(n).

   (a) $T(n) = 8T(n/2) + n^2$

   (b) $T(n) = 5T(n/2) + (n \log n)^2$

   (c) $T(n) = 2T(n/2) + n \log n$

   (d) $T(n) = T(n/2) + n$

   (e) $T(n) = 3T(n-1) - T(n-2)$