University of Nevada, Las Vegas Computer Science 477/677 Spring 2019

Assignment 3: Due Wednesday March 6, 2019

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets. Turn in the pages to the graduate assistant at the beginning of class, March 6.

1. Construct a treap with alphabetic key and numeric min-heap order. You are to insert the items one at a time and show the treap after each rotation. Insert letters in this order: D, N, L, H, J, K. The numeric heap keys (the random numbers) are given in the following table.

D	23
N	12
L	10
H	15
J	20
K	8

2. Solve each of the following recurrences, expressing the answers using O, Θ, or Ω, whichever is most appropriate.

   (a) $F(n) = F(n/2) + 1$

   (b) $F(n) = F(n-1) + O(n)$

   (c) $T(n) = 3T(n/2) + n^2$

(d) $F(n) = F(n - 2\sqrt{n} + 1) + n$ (Hint: Try a few values of n. I will give a more helpful clue later.)

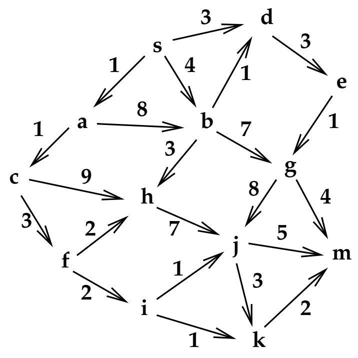
(e) $H(n) = H(n - \sqrt{n}) + n$ (Hint: Compare with Problem 2d.)

(f) This one is much harder. There is a certain well-known algorithm, which we will cover later, whose time complexity is given by the recurrence: $T(n) \leq T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + O(n)$, and by the condition $T(n) = \Omega(n)$. Express $T(n)$ using Θ notation.

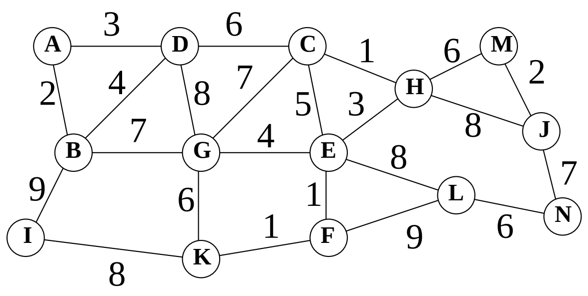
3. Use Huffman's algorithm to construct an optimal prefix code for the alphabet $\{A, B, C, D, E, F\}$ where the frequencies of the symbols are given by the following table.

A	6
B	4
C	5
D	8
E	14
F	3

4. Write the nodes of the following directed graph in topological order. (There could be more than one correct answer.)



5. Find a minimum spanning tree of the weighted graph shown below.



6. Insert the letters M, H, J, L, N, Q into an AVL tree in that order. Show the rotations (if any) after each insertion.

7. The divide and conquer algorithm, binary search, finds the median of a sorted array of length n is $O(\log n)$ time. Suppose a set of items is given to you in two sorted arrays, say X of length n and Y of length m. Write a divide and conquer algorithm which finds the median item of that set, namely the union of the items in X and the items in Y.

   For example, if you are given the arrays: $X = (1, 2, 5, 6, 9)$ and $Y = (3, 7, 10, 11)$, the median item is 6. You may assume that there are no duplicate items, if that helps. If the number of items is even, the median item could be either of the two middle items. I have not given this algorithm in class.