Homework: **Graph ADT's: Binary Heaps**

1. Check the Assignment Schedule for the DUE date.
2. Submit via Moodle.

Problem 1:

Suppose you built the MIN-HEAP for a graph with the node weights corresponding to the node degree in the graph. The root of the binary heap has the node with the MINIMUM degree in the graph, hence, MIN-HEAP. (Note: node and vertex are used interchangeably in this course.)

1. What is the cost of finding the node with the smallest degree using this ADT?
2. What is the cost of finding the node with the largest degree using this ADT? Justify your answer.
3. Suppose a high authority figure (a customer) from the U.S. CDC (Center for Disease Control) requests you to write a simulation program for estimating the spread (the number of infected people) of the virus propagation before CDC runs out of the vaccines. CDC will provide you the information about the social network (SN) of people who could be in direct contact. According to CDC, the number K of available vaccines is limitted, namely K << N, where N is the number of people in SN. **Your immunization strategy for selecting which people to immunize is based on the degree of the node in the SN**. At any time step of the simulation, only one person can be immunized. Once the person is rendered the vaccine, he/she can not infect the people he/she could be in touch with according to SN information.
    a. Given your current MIN-HEAP ADT implementation (C code is given in lecture notes on binary heaps), write the most efficient pseudo-code for this simulation.
    b. Clearly articulate all the assumptions of your model.
    c. Provide big-O analysis of time complexity of your simulation.
    d. Discuss pros and cons of your simulation program.

    Note: there is no correct/incorrect answer to this problem. The purpose is to give you an opportunity to experience the complexity of solving a realistic problem and to assess the reasoning you provide to justify your approach to this problem.

Problem 2: This problem is related to the previous problem.

1. What is the time complexity of finding the highest-degree vertex in a network, assuming the vertices are given to you in no specific order?
2. Assuming the network is stored in adjacency list format and the naive strategy is used to calculate the vertex of the highest degree, what is the time complexity of the entire process of finding the highest degree node and removing it from the network till no more vertices are left to remove. With the naive, or brute-force, strategy, the vertex with the highest degree is removed from consideration (i.e., after the person was immunized) and the search for the next highest degree node starts from scratch.

3. Given MIN-HEAP, how do you need to modify your implementation to solve the problem of removing the highest degree node till no more vertices are left? What is the time complexity of the entire calculation given these modifications?
4. The degree of the vertices in a simple graph are integers between zero and (N-1). It is possible to sort such a set of integers, in either increasing or decreasing manner, in time O(N). Describe an algorithm that achieves such a performance.