CSCI 570 - Fall 2016 - HW4

1. Design a data structure that has the following properties (assume n elements in the data structure, and that the data structure properties need to be preserved at the end of each operation):
   - Find median takes O(1) time
   - Extract-Median takes O(log n) time
   - Insert takes O(log n) time
   - Delete takes O(log n) time

   Do the following:

   a) Describe how your data structure will work.
   b) Give algorithms that implement the Extract-Median() and Insert() functions.

   Hint: Read this only if you really need to. Your Data Structure should use a min-heap and a max-heap simultaneously where half of the elements are in the max-heap and the other half are in the min-heap.

2. There is a stream of integers that comes continuously to a small server. The job of the server is to keep track of k largest numbers that it has seen so far. The server has the following restrictions:

   a) It can process only one number from the stream at a time, which means it takes a number from the stream, processes it, finishes with that number and takes the next number from the stream. It cannot take more than one number from the stream at a time due to memory restriction.
   b) It has enough memory to store up to k integers in a simple data structure (*e.g.* an array), and some extra memory for computation (like comparison, *etc.*).
   c) The time complexity for processing *one number* must be better than $\Theta(k)$. Anything that is $\Theta(k)$ or worse is not acceptable.

   Design an algorithm on the server to perform its job with the requirements listed above.

3. Consider the following modification to Dijkstra's algorithm for single source shortest paths to make it applicable to directed graphs with negative edge lengths. If the minimum edge length in the graph is $-w < 0$, then add $w + 1$ to each edge length thereby making all the edge lengths positive. Now apply Dijkstra's algorithm starting from the source s and output the

shortest paths to every other vertex. Does this modification work? Either prove that it correctly finds the shortest path starting from s to every vertex or give a counter example where it fails.

4. You are given a weighted directed graph G = (V, E, w) and the shortest path distances $\delta(s, u)$ from a source vertex s to every other vertex in G. However, you are not given $\pi(u)$ (the predecessor pointers). With this information, give an algorithm to find a shortest path from s to a given vertex t in $O(|V| + |E|)$ time.

5. We are given a directed graph G = (V, E) on which each edge $(u, v) \in E$ has an associated value r(u, v), which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret r(u, v) as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between any two given vertices

6. Consider two positively weighted graphs G = (V, E, w) and $G' = (V, E, w')$ with the same vertices V and edges E such that, for any edge $e \in E$, we have $w'(e) = w^2(e)$.

   a) Prove or disprove: For any two vertices $u, v \in V$, any shortest path between u and v in G' is also a shortest path in G.
   b) Prove or disprove: If T is a minimum spanning tree of G, then it must also be a minimum spanning tree for G'.

7. Assume that you are given a graph G and a minimum spanning tree T of G. A new edge e is added to G (without introducing any other vertices) to create a new graph $\bar{G}$. Design an algorithm that given G, T and e, finds a minimum spanning for $\bar{G}$. Your algorithm should run in O(n) time.

8. Solve Kleinberg and Tardos, Chapter 4, Exercise 8.

9. Solve Kleinberg and Tardos, Chapter 4, Exercise 21

10. Solve Kleinberg and Tardos, Chapter 4, Exercise 22.