CSCI 570 - Fall 2021 - HW 4

Due: Sep 23rd

Section 1: Heaps

Reading Assignment: Kleinberg and Tardos, Chapter 2.5.

Problem 1

[**10 points**] 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 $\mathcal{O}(1)$ time
- Insert takes $\mathcal{O}(\log n)$ time

Do the following:

1. Describe how your data structure will work.
2. Give algorithms that implement the Find-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.)

Problem 2

[**10 points**] 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:

- 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.
- 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.).

- The time complexity for processing one number must be better than $\mathcal{O}(k)$. Anything that is $\mathcal{O}(k)$ or worse is not acceptable.

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

Section 2: MST

Reading Assignment: Kleinberg and Tardos, Chapter 4.5.

Problem 3

[**10 points**] Suppose you are given a connected graph G, with edge costs that are all distinct. Prove that G has a unique minimum spanning tree.

Problem 4

[**10 points**] Let us say that a graph $G = (V, E)$ is a near tree if it is connected and has at most $n+8$ edges, where $n = |V|$. Give an algorithm with running time $\mathcal{O}(n)$ that takes a near tree G with costs on its edges, and returns a minimum spanning tree of G. You may assume that all edge costs are distinct.

Section 3: Shortest Path

Reading Assignment: Kleinberg and Tardos, Chapter 4.4.

Problem 5

[**20 points**] Given a connected graph $G = (V, E)$ with positive edge weights. In V, s and t are two nodes for shortest path computation, prove or disprove with explanations:

1. If all edge weights are unique, then there is a single shortest path between any two nodes in V.

2. If each edge's weight is increased by k, the shortest path cost between s and t will increase by a multiple of k.

3. If the weight of some edge e decreases by k, then the shortest path cost between s and t will decrease by at most k.

4. If each edge's weight is replaced by its square, i.e., w to w^2, then the shortest path between s and t will be the same as before but with different costs.

Problem 6

[**16 points**] Consider a directed, weighted graph G where all edge weights are positive. You have one Star, which allows you to change the weight of any one edge to zero. In other words, you may change the weight of any one edge to zero. Propose an efficient method based on Dijkstra's algorithm to find a lowest-cost path from node s to node t, given that you may set one edge weight to zero.

Note: you will receive 10 points if your algorithm is efficient. This means your method must do better than the naive solution, where the weight of each node is set to 0 per time and the Dijkstra's algorithm is applied every time for lowest-cost path searching. You will receive full points (16 points) if your algorithm has the same run time complexity as Dijkstra's algorithm.