CSCI 570 - Fall 2022 - HW 4

Due: September 21, 2022

Graded Problems

Problem 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 $\mathcal{O}(1)$ time

- Insert takes $\mathcal{O}(logn)$ time

Do the following:

1. Describe how your data structure will work.

2. Give algorithms that implement the Find-Median() and Insert() functions.

Problem 2

Let us say that a graph $G = (V, E)$ is a near tree if it is connected and has most $n + k$ edges, where $n = |V|$ and k is a **constant**. 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.

Problem 3

You are given a minimum spanning tree T in a graph $G = (V, E)$. Suppose we remove an edge from G creating a new graph G_1. Assuming that G_1 is still connected, devise a linear time algorithm to find a MST in G_1.

Problem 4

Considering the following graph G:
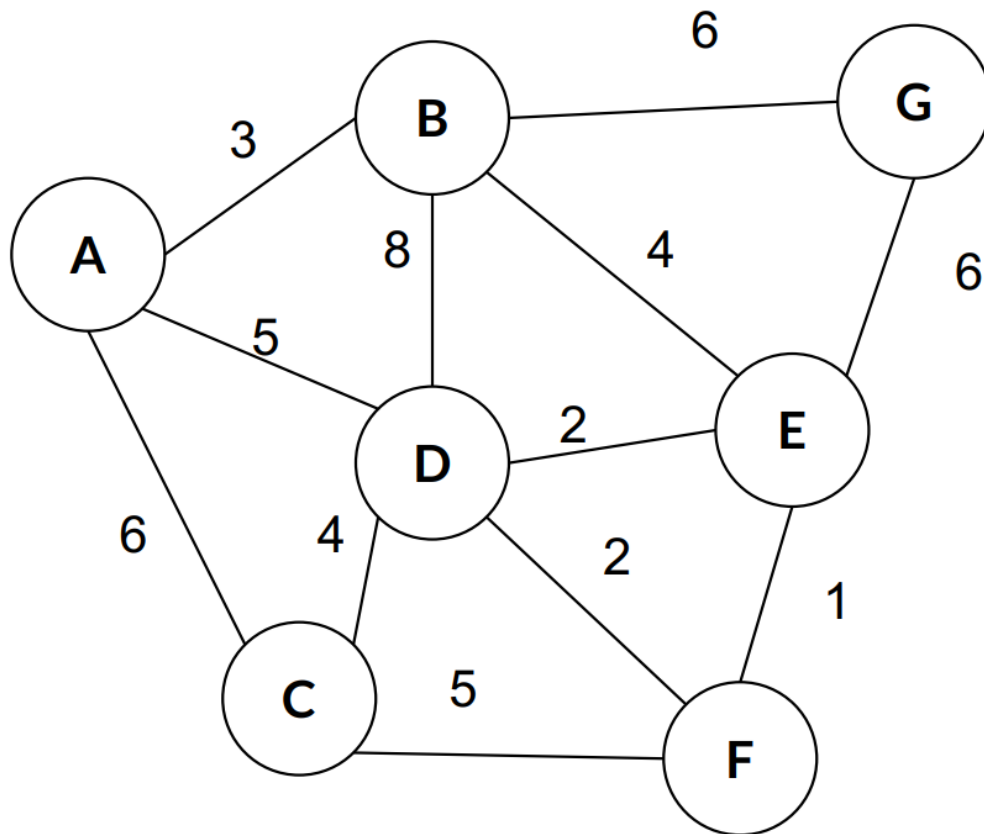
1. In graph G, if we use Kruskal's Algorithm to find the MST, what is the third edge added to the solution? Select all correct answers

    a. E-F

    b. D-E

    c. A-B

    d. C-F

    e. D-F

2. In graph G, if we use Prim's Algorithm to find MST starting at A, what is the second edge added to the solution?

    a. B-G

    b. B-E

c. D-E

d. A-D

e. E-F

3. What is the cost of the MST in the Graph?

a. 18

b. 19

c. 20

d. 21

e. 22

Problem 5

A new startup FastRoute wants to route information along a path in a communication network, represented as a graph. Each vertex represents a router and each edge a wire between routers. The wires are weighted by the maximum bandwidth they can support. FastRoute comes to you and asks you to develop an algorithm to find the path with maximum bandwidth from any source s to any destination t. As you would expect, the bandwidth of a path is the minimum of the bandwidths of the edges on that path; the minimum edge is the bottleneck. Explain how to modify Dijkstra's algorithm to do this.

Problem 6

A network of n servers under your supervision is modeled as an undirected graph $G = (V, E)$ where a vertex in the graph corresponds to a server in the network and an edge models a link between the two servers corresponding to its incident vertices. Assume G is connected. Each edge is labeled with a positive integer that represents the cost of maintaining the link it models. Further, there is one server (call its corresponding vertex as S) that is not reliable and likely to fail. Due to a budget cut, you decide to remove a subset of the links while still ensuring connectivity. That is, you decide to remove a subset of E so that the remaining graph is a spanning tree. Further, to ensure that the failure of S does not affect the rest of the network, you also require that S is connected to exactly one other vertex in the remaining graph. Design an algorithm that given G and the edge costs efficiently decides if it is possible to remove a subset of E, such that the remaining graph is a spanning tree where S is connected to exactly one other vertex and (if possible) finds a solution that minimizes the sum of maintenance costs of the remaining edges.

Problem 7

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 8

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.