CSCI 570 - Summer 2023 - HW 3

Due: July 16, 2023

Graded Problems

Problem 1

[5 points] 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.

Solution

Suppose the removed edge is named e_r.

Case 1: If e_r is not in T. The MST is T. This takes linear time.

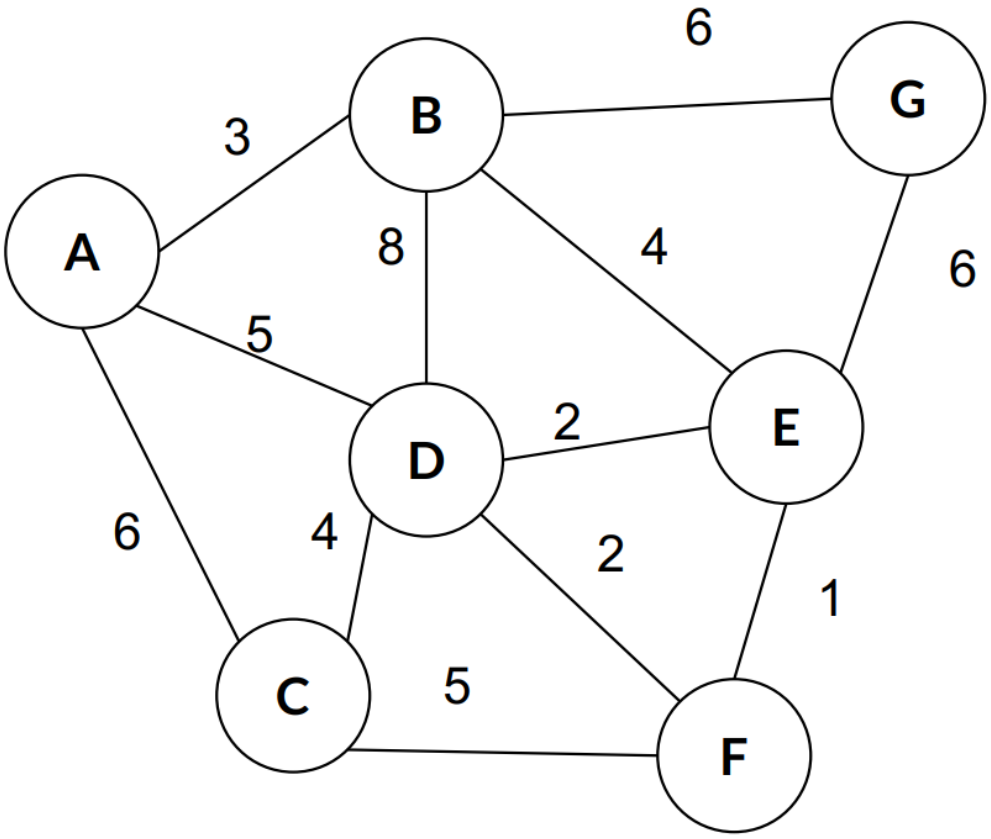
Case 2: If e_r is in T. Once we delete an edge from T, the tree becomes disconnected. We need to find the minimum weight edge that connects two components, T_1 and T_2. Pick any component say T_2 and find all edges going to T_1. Among them choose the one which has the smallest cost. Runtime $\mathcal{O}(E)$.

Rubric

- 2 pt: The first case
- 3 pt: The second case

Problem 2

[15 points] 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

Solution

1. c. A-B

2. b. B-E

3. c. 20

Rubric

5 points each

Problem 3

[20 points] 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.

Solution

Data structure change: we'll use a max heap instead of a min heap used in Dijkstra's.

Initialization of the heap. Initially all nodes will have a distance (bandwidth) of zero from s, except the starting point s which will have a bandwidth of ∞ to itself.

Change in relaxation step. Based on the definition of a path's bandwidth, the bandwidth of a path from s to u through u's neighbor v will be $d(u) = \min(d(v), weight(v, u))$, because

the bandwidth of a path is equal to the bandwidth of its lowest bandwidth edge. Therefore, in the relaxation step, we will be replacing:

$d(u) = \min(d(u), d(v) + weight(v, u))$

with

$d(u) = \max(d(u), \min(d(v), weight(v, u))$

Rubric

- 5 points for max heap
- 5 points for initialization
- 10 points for relaxation modification

Problem 4

[20 points] 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.

Solution

First we need to check the possibility of node S having only one neighbor in a spanning tree of the underlying graph. The best way to check this is to remove node S and all its adjacent edges to form a graph $G\prime$. If $G\prime$ is a connected graph, then we can claim it is possible to have a spanning tree where node S has only one neighbor. To check the connectivity of $G\prime$, the simplest way is to run a DFS or BFS algorithm on $G\prime$.

Considering that $G\prime$ is connected, we need to find the spanning tree than minimizes the maintenance cost. Therefore we need to find the MST with the additional constraint that S should be a leaf. Therefore, we remove S and all its adjacent edges to form $G\prime$. We run Prime's (or any other MST algorithm) to find the MST of $G\prime$. Among all edges adjacent to S, we find the one with the minimum maintenance cost and connect S to the MST using

this edge. The resulting graph will still be a spanning tree and in this spanning tree S will be a leaf.

Rubric

- 10 points: Building your spanning tree in a way that node S is a leaf
- 10 points: Making sure that the final MST is connected

Problem 5

[20 points] Prove or disprove the following:

- T is a spanning tree on an undirected graph $G = (V, E)$. Edge costs in G are NOT guaranteed to be unique. If every edge in T belongs to SOME minimum cost spanning trees in G, then T is itself a minimum cost spanning tree.
- 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 $w'(e) = w(e)^2$ For any two vertices u, v in V , any shortest path between u and v in G' is also a shortest path in G.

Solution

1. False. Counter example:
   T = ab, bc
   MST1 = ab, ac
   MST2 = ac, bc
   Ab is in MST1
   Bc is in MST2
   However, T is not a MST

   Rubric
   - 4 pt: Correct T/F claim
   - 6 pt: Provides a correct counterexample as explanation

2. False. Assume we have two paths in G, one with weights 2 and 2 and another one with weight 3. The first one is shorter in G' while the second one is shorter in G.

   Rubric
   - 4 pt: Correct T/F claim
   - 6 pt: Provides a correct counterexample as explanation

Problem 6

[20 points] You have been tasked with organizing a conference that will bring together participants from different countries, each with their own native language. In order to facilitate communication and ensure that all participants can understand each other, you need to hire a set of interpreters. However, due to budget constraints, you can only hire a limited number of interpreters. There are n different languages spoken by the participants and m available interpreters. Each interpreter is fluent in exactly two languages and can provide simultaneous interpretation between them. Each interpreter has a specific hiring cost associated with them. Your goal is to determine the minimum cost of hiring a subset of interpreters such that every participant can understand each other, either directly or through a chain of interpreters. Design an efficient algorithm to solve this problem and find the minimum cost of hiring interpreters for the conference.

G: Graph representing the interpreters and their language pairs
V: Set of vertices representing the languages
E: Set of edges representing the language pairs between interpreters
n: Number of different languages spoken by the participants
m: Number of available interpreters

Solution To solve the problem of hiring interpreters while minimizing the cost using a minimum spanning tree (MST) approach, we can modify the original graph representation and apply an MST algorithm:
Create a graph G with vertices V representing the languages and edges E representing the interpreters pairs between languages.
Assign a weight to each edge in E representing the hiring cost of an interpreter fluent in the corresponding language pair.
Find the minimum spanning tree T of the graph G using a suitable MST algorithm like Kruskal's algorithm or Prim's algorithm. This MST will represent the minimum cost hiring plan for the interpreters.
Construct a new graph G' with vertices V' representing the interpreters from the MST T and edges E' representing the language pairs between these interpreters.
Calculate the total cost of hiring interpreters in G' by summing up the weights of all the edges in E'. Return the minimum cost obtained from step 5 as the optimal solution for hiring interpreters.
The time complexity of this algorithm depends on the chosen MST algorithm. Kruskal's algorithm typically has a time complexity of $O(|E|log|V|)$, while Prim's algorithm has a time complexity of $O(|V|^2)$ with a binary heap implementation. The space complexity is $O(|V| + |E|)$ to store the graph representation.

Rubric For the algorithm:

- 5 pt if the approach is finding the MST

- 5 pt MST reconstruct on is correct.
- 10 pt if the algorithm steps are correct and with efficient time complexity.