CS3510-A: Design and Analysis of Algorithms, Spring 2021

Homework-7

March 2, 2021

DUE DATE: Tuesday, March 9, 11:59pm

Note-1: Your homework solutions should be electronically formatted as a single PDF document that you will upload on Gradescope. If you have to include some handwritten parts, please make sure that they are very clearly written and that you include them as high resolution images.
Note-2: Please think twice before you copy a solution from another student or resource (book, web site, etc). It is not worth the risk and embarrassment.
Note-3: You need to **explain/justify** your answers. Do not expect full credit if you just state the correct answer. This includes their correctness and runtime.
Note-4: You will get 2 extra points if you submit electronically typed solutions instead of hand-written.

Problem-1 (30 points)

We are given a directed graph with positive link costs. Design an algorithm that computes the minimum-cost cycle in the graph. If the graph is acyclic, the algorithm should report 0. The algorithm should be $O(n\,X)$, where $O(X)$ is Dijkstra's shortest path algorithm runtime complexity and n is the number of vertices in the graph. **Show the runtime in terms of m and n** (do not use X; we use it so as to not give away the runtime complexity).

Note: You must show your algorithm's running time in terms of n and m.

Solution

Problem-2 (35 points)

Suppose you're a consultant for the networking company CluNet, and they have the following problem. The network that they're currently working on is modeled by a connected graph $G = (V, E)$ with n nodes. Each edge e is a fiber-optic cable that is owned by one of two companies – creatively named X and Y – and leased to CluNet.

Their plan is to choose a spanning tree T of G and upgrade the links corresponding to the edges of T. Their business relations people have already concluded an agreement with companies X and Y stipulating a number k so that in the tree T that is chosen, k of the edges will be owned by X and $n - k - 1$ of the edges will be owned by Y.

CluNet management now faces the following problem. It is not at all clear to them whether there even *exists* a spanning tree T meeting these conditions, or how to find one if it exists. So this is the problem they put to you: Give a polynomial-time algorithm that takes G, with each edge labeled X and Y, and either (i) returns a spanning tree with exactly k edges labeled X, or (ii) reports correctly that no such tree exists.

Solution

Problem-3 (35 points)

Suppose we are given a connected graph $G = (V, E)$ with $|V| = n, |E| = m$. Each edge $e \in E$ has a positive weight $w(e) > 0$. Each vertex $v \in V$ has a label $a_v \in \{1, 2, \ldots, k\}$. Each label $i \in \{1, 2, \ldots, k\}$, has a cost c_i. At each vertex $u \in V$, you can perform two types of operations:

1. You can travel from u to some vertex v in its neighborhood, i.e. any v such that $e = (u, v) \in E$. The cost of this operation is $w(e)$.

2. You can travel from u to some vertex v with the same label, i.e. any v such that $a_u = a_v$. The cost of this operation is c_{a_u}.

Given a source vertex s and a sink vertex t, design a $O(X)$ algorithm that computes the minimal cost of traveling from s to t where $O(X)$ is Dijkstra's shortest path algorithm runtime complexity on a graph with $O(n)$ vertices and $O(m)$ edges.

Solution