Problem Set 10

Data Structures and Algorithms, Fall 2021

Due: December 9, in class.

Problem 1

Given a directed acyclic graph $G = (V, E)$ and two vertices s and t in V, devise an algorithm that returns the number of simple paths from s to t in G. (Your algorithm needs only to count the simple paths, not list them.) You need to give the pseudocode of your algorithm, and your algorithm should have runtime $O(|V| + |E|)$.

Problem 2

Given a directed graph $G = (V, E)$, devise an algorithm to create another graph $G' = (V, E')$ such that: (a) G' has the same strongly connected components as G, (b) G' has the same component graph as G, and (c) E' is as small as possible. Your algorithm needs to return G' and have runtime $O(|V| + |E|)$. You also need to give the pseudocode of your algorithm.

Problem 3

The police department in the city of Computopia has made all streets one-way. The mayor claims there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. However, the city elections are coming up soon, and there is just enough time to run a linear-time algorithm.

(a) Formulate this problem graph-theoretically, and explain why it can be solved in linear time.

(b) Suppose it now turns out that the mayor's original claim is false. She next claims something weaker: if you start driving from town hall, navigating one-way streets, then no matter where you reach, there is always a way to drive legally back to the town hall. Formulate this weaker property as a graph-theoretic problem, and show how it too can be checked in linear time.

Problem 4

In the 2SAT problem, you are given a set of *clauses*, where each clause is the disjunction (OR) of two *literals* (a literal is a Boolean variable or the negation of a Boolean variable). You are looking for a way to assign a value `true` or `false` to each of the variables so that all clauses are satisfied — that is, there is at least one true literal in each clause. For example, here's an instance of 2SAT:

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_4)$$

This instance has a satisfying assignment: set x_1, x_2, x_3, and x_4 to `true`, `false`, `false`, and `true`, respectively. The purpose of this problem is to lead you to a way of solving 2SAT efficiently by reducing it to the problem of finding the strongly connected components of a directed graph. Given an instance I of 2SAT with n variables and m clauses, construct a directed graph $G_I = (V, E)$ as follows.

- G_I has $2n$ nodes, one for each variable and its negation.
- G_I has $2m$ edges: for each clause $(\alpha \vee \beta)$ of I (where α, β are literals), G_I has an edge from the negation of α to β, and one from the negation of β to α.

(a) Carry out this construction for the instance of 2SAT given above.

(b) Show that if G_I has a strongly connected component containing both x and $\overline{x}$ for some variable x, then I has no satisfying assignment.

(c) Now show the converse of (b): if none of G_I's strongly connected components contain both a literal and its negation, then the instance I must be satisfiable.

(d) Describe a linear-time (that is, $O(n+m)$ time) algorithm for solving 2SAT.

Problem 5

(a) Prove that if all its edge weights of an undirected graph G are distinct, then G has a unique minimum spanning tree.

(b) Describe an edge-weighted undirected graph that has a unique minimum spanning tree, even though two edges have equal weights.

(c) Prove or disprove: The minimum spanning tree of an undirected graph G includes the minimum-weight edge in every cycle in G.

Problem 6

A *feedback edge set* of an undirected graph G is a subset F of the edges such that every cycle in G contains at least one edge in F. In other words, removing every edge in F makes the graph G acyclic. Describe and analyze a fast algorithm to compute the minimum-weight feedback edge set of a given edge-weighted graph. *(Hint: how to compute a maximum spanning tree?)*

Problem 7

Suppose we are given both an undirected graph $G = (V, E)$ with weighted edges and a minimum spanning tree $T = (V, E')$ of G.

(a) Describe an algorithm to update the minimum spanning tree when the weight of a single edge $e \in E$ is decreased. Remember to analyze the runtime of your algorithm.

(b) Describe an algorithm to update the minimum spanning tree when the weight of a single edge $e \in E$ is increased. Remember to analyze the runtime of your algorithm.

In both cases, the input to your algorithm is the edge e and its new weight. Your algorithms should modify T so that it is still a minimum spanning tree. You should make your algorithm as fast as possible. *(Hint: Consider the cases $e \in E'$ and $e \notin E'$ separately.)*