CS 577: Introduction to Algorithms — Homework 3

Out: 02/10/16 — **Due: 02/17/16**

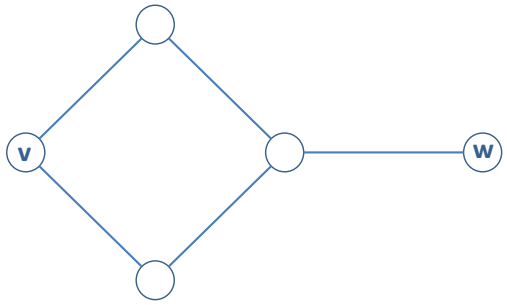
Ground Rules

See HW1.

Problems

1. (Restatement of problem 3.10 in the book.) Sometimes there are multiple shortest paths between pairs of nodes in a graph. Develop an algorithm for the following task: given an undirected graph $G = (V, E)$ with unit edge lengths and nodes v and w, output the number of distinct shortest paths from v to w. For example, for the graph below, on input v and w your algorithm should output 2. Your algorithm should run in linear time. Prove the correctness of your algorithm and analyze its running time.

   Hint: Think about modifying BFS.



2. (Restatement of problem 4.18 in the book.) You are given a directed graph $G = (V, E)$ with a function $f_e(t)$ for every edge $e \in E$ that specifies how long it takes to "walk" along that edge. In particular, for any edge $e = (u \to v)$, if you walk along the edge starting at u at time t, then you arrive at v at time $f_e(t)$. The functions $f_e(t)$ have the following properties: (1) $f_e(t) \geq t$ for all $e \in E$ and $t \geq 0$, that is, you cannot travel back in time; (2) $f_e(t)$ is an increasing function of t, that is, a later start cannot lead to an earlier arrival.

   With this given information, you are required to develop a shortest paths algorithm for this graph. Specifically, given two nodes s and t, your algorithm should return a path from s to t so that starting at s at time 0 and following that path gets you to t as early as possible. Your algorithm should run in time polynomial in the size of the graph; you may assume that arithmetic operations take unit time.

   Prove the correctness of your algorithm and analyze its running time.