CS 577: Introduction to Algorithms	Homework 5
Out: 02/24/17	**Due: 03/03/17**

Ground Rules

- Review problems will be discussed at the following week's review.
- Self-graded problems will be discussed at the following week's review, and are to be self-graded by you during the review. Only students present at the review session and turning in their graded solutions in person will earn points.
- Graded problems are to be turned in on the due date at the beginning of the lecture.
- Self-graded problems should be done individually.
- Graded problems should be done in pairs.
- Please follow page limits for all self-graded and graded problems.
- Write your name(s) clearly on your submissions, and turn in each problem on **a separate sheet of paper**.

Review Problems

1. In class we saw that to prove the correctness of Dijkstra's algorithm we need to assert that all edges weights in the graph are nonnegative. In this problem we examine this assumption closely.
   (a) Consider the following algorithm for finding shortest paths in graphs with negative edge weights: add a large constant to each edge weight so that all the weights become positive; then run Dijkstras algorithm. Is this algorithm correct? Prove your answer.
   (b) Suppose that the source node s has no incoming edges, and all of the negative weight edges are those that leave s (so, in particular, there are no negative cost cycles). Can Dijkstra's algorithm, started at s, fail on such a graph? Prove your answer.
2. Let us say that a graph $G = (V, E)$ is a near-tree if it is connected and has at most $n + 10$ edges, where $n = |V|$. Give an algorithm with running time $O(n)$ that takes a near-tree G with costs on its edges, and returns a minimum spanning tree of G. You may assume that all the edge costs are distinct.

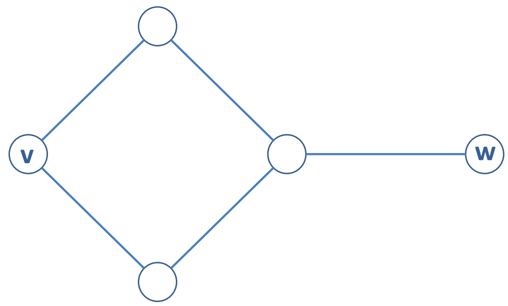
Self-graded Problems

3. **(Page limit: 1 sheet; 2 sides)** You are given a directed acyclic graph with (positive or negative) lengths on edges. Design a polynomial time algorithm to find the longest directed path in this graph. The path may start and end at any pair of distinct nodes.

Graded Problems

4. **(Page limit: 1 sheet; 2 sides)** 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.



5. **(Page limit: 1 sheet; 2 sides)** Solve Problem 12, **Racetrack**, on Page 12-13 of Chapter 18 in the book "Algorithms, Etc."

   Link: `http://jeffe.cs.illinois.edu/teaching/algorithms/notes/18-graphs.pdf.`