Algorithms, Winter 2010-11, Homework 6
due Friday 2/4/11, 4:00pm

For every problem include the pseudocode of your algorithm and a short verbal description. Briefly argue your algorithm's correctness and explain its running time. Submit your code for every problem.

Problem 1

Give a linear-time algorithm (i.e., $O(n+m)$) that decides whether a given undirected graph is cyclic.

Problem 2

Give an $O(n^2)$ algorithm that computes the length of the longest path in a given directed acyclic graph.

Hint: use topological sort and dynamic programming.

Note: the problem of computing the longest path in undirected graphs or directed graphs with cycles, is known to be NP-complete. This means that we do not know whether a polynomial-time algorithm exists or not (most people think that it does not exist). However, if the input is a directed acyclic graph, this problem can be solved in polynomial-time.

Problem 3

Given is a weighted undirected graph $G = (V, E)$ with positive weights and a subset of its edges $F \subseteq E$. An *F-containing spanning tree of G* is a spanning tree that contains all edges from F (there might be other edges as well). Give an $O(mn)$ algorithm that finds the cost of the minimum-cost F-containing spanning tree of G.