CSE332 Data Abstractions, Spring 2010
Homework 5

Due: **Friday, May 14, 2010** at the beginning of class. Your work should be readable as well as correct.

This assignment has **four** problems.

Problem 1. Graph Representation

Suppose a directed graph has a million nodes, most nodes have only a few edges, but a few nodes have hundreds of thousands of edges.

- In what way(s) would an adjacency-matrix representation of this graph lead to inefficiencies?
- In what way(s) would an adjacency-list representation of this graph lead to inefficiencies?
- Design a representation for this sort of graph that avoids all the inefficiences in your answers to parts (a) and (b).
- *Optional question:* Can you think of any situations where this sort of "unbalanced" graph might arise?

Problem 2. Topological Sort

Weiss, problem 9.1. For each step, show the in-degree array and the queue.

Problem 3. How To Graduate As Soon As Possible

(a) Given a DAG representing course pre-requisites, use precise English to describe an algorithm for computing the minimum number of academic terms that it would take to complete all the courses. Assume that there is no limit on how many courses you can take in any given term and that every course is offered every term.

(b) Explain how to extend your algorithm slightly to produce a course schedule such that all courses are taken in the minimum number of academic terms.

(c) What is the asymptotic running time of your algorithm in terms of $|V|$ and $|E|$?

Problem 4. Dijkstra's Algorithm

(a) Weiss, problem 9.5(a). Use Dijkstra's algorithm and show the results of the algorithm in the form used in lecture — a table showing for each vertex its best-known distance from the starting vertex and its predecessor vertex on the path. Also show the order in which the vertices are added to the "cloud" of known vertices as the algorithm progresses.

(b) If there is more than one minimum cost path from v to w, will Dijkstra's algorithm always find the path with the fewest edges? If not, explain in a few sentences how to modify Dijkstra's algorithm so that if there is more than one minimum path from v to w, a path with the fewest edges is chosen.

(c) Give an example where Dijkstra's algorithm gives the wrong answer in the presence of a negative-cost edge but no negative-cost cycles. Explain why Dijkstra's algorithm fails on your example.

(d) Suppose you are given a graph that has negative-cost edges but no negative-cost cycles. Consider the following strategy to find shortest paths in this graph: uniformly add a constant k to the cost of every edge, so that all costs become non-negative, then run Dijkstra's algorithm and return that result with the edge costs reverted back to their original values (i.e., with k subtracted). Give an example where this technique fails and explain why it does so. (Hint: one simple example uses only three vertices.) Also, give a general explanation as to why this technique does not work.