CSE 332 Data Abstractions, Winter 2012

Homework 5

Due Friday, Feb. 17th, 2012 at the beginning of lecture. Please be sure your work is readable (either written clearly or typed).

Problem 1. Graph Representation

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

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

Problem 2. Topological Sort

Weiss, problem 9.1 (the problem is the same in the 2nd and 3rd editions of the textbook). For each step, show the in-degree array and the queue.

Problem 3. Dijkstra's Algorithm

a. Weiss, problem 9.5(a) (the problem is the same in the 2nd and 3rd editions of the textbook). 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; you can use a single table and cross-out/add values as you progress. Also write and circle the order in which the vertices are marked as visited.
b. If there is more than one minimum cost path from vertices 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. The example need not be complex; it is possible to demonstrate the point using as few as 3 vertices.
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.