Algorithms Section 2
Spring 2003
Homework 7a
Due Thursday, March 13

1. **[15 Points] Detecting Cycles!**

   (a) Consider an **undirected graph** with V vertices and E edges. The graph may be disconnected. That is, the graph may consist of a number of seperate disconnected components. Describe how depth-first search can be used to determine whether or not a graph has a cycle in $O(V+E)$ time. (That is, whether at least one of its components has a cycle.) You should assume that the graph is represented by an adjacency list. Explain your algorithm and your analysis carefully. This problem requires a bit of cleverness in the data structure in order to get the $O(V+E)$ time!

   (b) Does your above algorithm work if you use breadth-first search instead of depth-first search? Explain briefly.

   (c) The algorithm that you described in the first part of this problem can easily be modified to work for directed graphs as well. However, now show how the topological sorting algorithm can be used to determine whether a directed graph contains a directed cycle. Again, the graph may be disconnected. The algorithm should work in $O(V+E)$ time assuming the graph is represented as an adjacency list. Explain your algorithm and your analysis carefully.

2. **[10 Points] Network Reliability!** Millisoft has decided to buy the entire Internet and rename it the Millinet. They've hired you to solve a particularly interesting routing problem in the Millinet. "Here's the story," says Gill Bates, propping his feet up on your desk. "The Millinet can be represented by an undirected graph where the vertices are the routers in the network and the edges represent network links between those routers. Each edge has a certain reliability, which is a real number between 0 and 1. The reliability is just the probability that the edge will be available for transmitting data. Since the reliabilities are statistically independent, the reliability of a path between two nodes is just the product of the reliabilities of the edges on that path. You will be given an $V \times V$ adjacency matrix such that the element at row i and column j of the matrix is the reliability of the edge between vertices i and j if it exists and is 0 otherwise. Your job is to find an algorithm to compute the most reliable path between all pairs of vertices."

   (a) Explain how the Floyd-Warshall Algorithm can be modified to compute the reliability of the most reliable path between every pair of vertices.

   (b) What is the running time of your algorithm? Why?