Algorithms
Computer Science 140 & Mathematics 168
Spring 2007
Homework 7a
Due Thursday, March 1

- We are now covering material from Chapter 4 in our book.
- This homework is quite light. Please try to finish it early and start studying for the exam.
- Exam 1 will be given out in class on Thursday, March 1 and will be due on Tuesday, March 6 at 5 PM in the CS office. The exam has no time limit and you need NOT take it in one contiguous sitting. Here's what you may use on the exam:
  - Your own homework sets.
  - The solution sets that were provided.
  - Class notes with your own markup on them. If you took notes on your laptop, you may use those.
  - Any additional notes that you prepare before the exam.

  No other materials may be used.
- You may send requests by e-mail for clarification on the wording of exam questions until Thursday at 11 PM. The questions will be answered by Thursday at midnight. Afterwards, there will be no communication about the exam. There will be no office hours on Friday, March 2 or Monday, March 5.

1. **[10 Points] Hurts Car Rental!** Hurts Car Rental provides a variety of maps and travel information for its customers. Customers often complain that certain travel routes are boring because they involve long stretches of road with no attractions in between. Hurts would like to advise travelers of the boringness of certain trips.

   Let G be an undirected graph with n vertices (representing cities, Spam factories, and other major attractions) and weighted edges (representing undirected roads and their lengths). The *boringness* of a path between vertices u and v in G is defined to be the maximum weight among all edges on that path. The *cob (coefficient of boringness)* of a pair (u, v) is the minimum boringness over all paths from u to v.

The weights (lengths of the roads between the vertices) are specified by a symmetric $n \times n$ adjacency matrix where the element in row i and column j specifies the weight of the edge between vertices i and j and is ∞ if no such edge exists.

(a) Describe a modification to the Floyd-Warshall Algorithm to compute the cobs of all pairs of vertices in the graph and very briefly explain why it is correct.

(b) Explain how the algorithm could be augmented to allow us to later find an actual least boring path between any two cities.

(c) What is the running time of your algorithm to compute the cobs? Explain briefly.

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 a symmetric $n \times n$ 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) Describe an algorithm that computes the reliability of the most reliable path between every pair of vertices and very briefly explain why it is correct.

(b) Briefly explain how the algorithm can be augmented to reconstruct the actual most reliable path.

(c) What is the running time of your algorithm to compute the reliabilities? Explain briefly.

(*Note:* There are at least two rather different ways of doing this problem, both of which result in correct solutions with the same running time. If you find one way, you might want to see if you can find another!)