Algorithms
Computer Science 140 & Mathematics 168
Instructor: B. Thom
Spring 2004
Homework 10a
Due on 4/1/04 (beginning of class)

1. **[30 Points] Hurts Car Rental and Optimal Boringness!**

   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 a directed graph with V vertices (representing cities, canned meat product museums, and other major attractions) and E directed edges (representing one-way roads) with positive edge weights representing the lengths of these roads. The *boringness value* of a path between vertices u and v in G is defined to be the edge of maximum weight on that path. The *optimal boringness* of a pair (u, v) is the minimum boringness value over all paths from u to v. In other words, for each path from u to v there is a boringness value. Imagine that we were to list the boringness values for all paths from u to v. The minimum of these values would be the optimal boringness of the pair (u, v). A Hurts customer would then travel on a path with optimal boringness.

   (a) Describe a $O(V^3)$ algorithm for computing the optimal boringness between all pairs of vertices. Remember, the graph is directed and the edge weights are positive.

   (b) Professor Nils Bohringness is interested in finding the optimal boringness from a given start vertex s to all other vertices in the graph. He has proposed the following modification to Dijkstra's Algorithm: Initially, label vertex s with the value 0 and label all other vertices ∞. At each iteration, choose the vertex u with minimum label and lock its value. This will be the optimal boringness from s to u. Now, vertex u relaxes all of its neighbors as follows: If there is an edge from u to v, vertex u offers vertex v the maximum of the label of u and the weight of the edge from u to v. Vertex v accepts this offer if it is less than its current label. This process is repeated until all vertices are locked.

   Prove that Professor Bohringness' Algorithm is correct by giving a very carefully and precisely written proof by induction. You should follow the general structure that was used in class when we proved Dijkstra's algorithm.