Algorithms Section 2
Spring 2003
Homework 7b
Due Tuesday, March 25

1. [**25 Points**] **Implementing the Floyd-Warshall Algorithm!** In this problem you will implement the Floyd-Warshall shortest path algorithm in your favorite programming language. For simplicity, you may assume that the graph will always have 5 vertices. The program will then find the shortest path between every pair of vertices. For each pair of vertices, i and j, it must print the distance from i to j and then the *actual shortest path* from i to j.

   To see an example of how your program should behave, go to the directory `~bthom/public_html/algorithms` on turing. The file `data` contains an adjacency matrix for the graph shown on page 626 of our textbook. Notice that the value 10000 is being used instead of ∞. You can run the one we implemented for demonstration purposes, called `floyd`, by using Unix redirection. Simply type `floyd < data`.

   You should turn in a printout of your source code as well a script showing your program running on the file `data` in Belinda's directory.

2. [**10 Points**] **Finding a Negative Cost Cycle.** In class we showed how the Bellman-Ford Algorithm can be used to detect if a negative cost cycle exists in a directed graph. Now explain how an actual negative cost cycle can be found (that is, the vertices and edges in one such cycle can be printed out) if one exists. Your algorithm needs to only find one such negative cost cycle. Describe the algorithm and analyze its running time. Your algorithm should run in time polynomial in V and E, but it's not required to be the fastest possible algorithm for this problem.