CSE 2320 Lab Assignment 4

Due August 12, 2016

Goal:

Understanding of the Floyd-Warshall algorithm with successors.

Requirements:

1. Write a C program to compute all-pairs shortest paths with an *odd* number of edges, along with those with an *even* number of edges, for a directed graph. These results are computed at the same time and are stored in matrices. The successor matrices must also be maintained.

   The graph will have no more than 10 vertices. The input format is the same as `floydWarshall.c` on the course web page. Edge weights will be non-negative values not exceeding 100.

   The input will be read from standard input. Do NOT prompt for a file name!

   The output is similar to `floydWarshall.c`:

   a. After each column is processed, output the matrices for the shortest paths with an odd number of edges and an even number of edges. Successor matrices should also be output.

   b. After the last column is processed, print whether or not each pair of vertices has a path with an odd number of edges. Likewise, print whether or not each pair of vertices has a path with an even number of edges. For the paths that exist, use the successor information from the two matrices to list the vertices on each path.

2. Submit your C program source file by 5:00 pm on August 12.

Getting Started:

1. The details regarding oddness and evenness may be developed by using the transitivity diagram from Notes 16, page 5.

2. Zero-edge, i.e. reflexive, paths are not to be considered. Entries in the adjacency matrices should be initialized with a distance of "infinity". It is acceptable for self-loops to appear in the input file. Input edges are only recorded in the "odd" matrix, i.e. the "even" matrix is initialized for no even paths existing.

3. A vertex may appear several times on a path so that "oddness" or "evenness" may be achieved. This also applies to the starting and destination vertices. Thus, the simple loop in `floydWarshall.c` for traversing the successors to output a path must be modified to account for paths with even and odd numbers of edges. It is easy to check this code by summing the edge lengths to compare with the matrix distance.