CSE 2320 Lab Assignment 4

Due August 12, 2015

Goals:

1. Understanding of hashing.

2. Understanding of graphs.

3. Understanding of strongly-connected components (Notes 14).

Requirements:

1. Design, code, and test a C program to determine strongly connected components for a directed graph. You may modify
    `http://ranger.uta.edu/~weems/NOTES2320/dfsSCC.c`

    a. Input is to be read from standard input (like the first three assignments):

        1. The first line is two integer values: `n`, the number of vertices, and `m`, the number of edges.
        2. The remaining `m` lines will each contain two values defining an edge: a tail name (string of no more than 25 characters) and a head name (another string).

    b. While reading the input, new vertex names should be stored in a double hash table along with consecutively assigned vertex numbers. The first line of your output should indicate the size of your double hash table.

        In addition to the double hash table, a table of vertex names will be needed. This is needed when printing your results for the end user.

        If the input does not have exactly `n` different names, give a disparaging message and stop.

        The assigned vertex numbers are used to build compressed adjacency lists.

    c. Perform Kosaraju's SCC algorithm (Notes 14). The elements of each SCC must be output using the vertex *names*, not numbers.

2. Submit your C program on Blackboard by 5:45 p.m. on August 12. One of the comment lines should include the compilation command used on OMEGA.

Getting Started:

1. Your double hash table should have the smallest prime size to assure a load factor no larger than 90%. You will need code for a simple function to compute this value.

2. The SCCs for a given graph are unique, but the output order could vary.