CSCI 232: Data Structures: Project 4

Due Thursday, Dec. 15 @ 11pm

Project Description

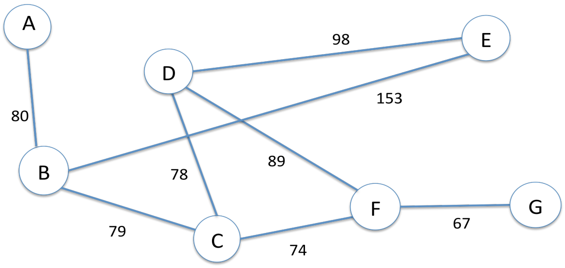
For this project, you are to develop a user interface that allows programmers to create graphs and test graph algorithms with them. Your program should allow the user to enter a description of a graph's vertices and edges as well as a starting vertex for certain tests. Your interface menu should also allow users to select among the following graph algorithms:

- Single-source shortest path
- All-pairs shortest path
- Topological sort

Some (or parts) of these algorithms have been provided for you in your text, but you need to complete them and package everything together in a modular way with a clear and simple user interface.

Note that for the shortest path problems, your algorithms should run on undirected weighted graphs. For the topological sort, you will need a directed graph as input. Therefore, your program should be able to read in and represent both types of graphs.

Consider the following undirected weighted graph. Below is an example of what a session with your program might look like using this graph.



Graph 1: Undirected weighted graph.

```
Main Menu
  1  Input a graph from the keyboard
  2  Input a graph from a file
  3  View graph
  4  Single-source shortest path
  5  All-pairs shortest path
  6  Topological sort
  7  Exit the program

Enter your selection[1-7]: 1

Enter an edge or return to quit: A-B: 80
Enter an edge or return to quit: B-C: 79
Enter an edge or return to quit: B-E: 153
Enter an edge or return to quit: C-D: 78
Enter an edge or return to quit: C-F: 74
Enter an edge or return to quit: D-E: 98
Enter an edge or return to quit: D-F: 89
Enter an edge or return to quit: F-G: 67
Enter an edge or return to quit: D-F: 89
Enter an edge or return to quit:
Graph created successfully

Main Menu
  1  Input a graph from the keyboard
  2  Input a graph from a file
  3  View graph
  4  Single-source shortest path
  5  All-pairs shortest path
  6  Topological sort
  7  Exit the program

Enter your selection[1-7]: 4
Enter the start vertex: B

The shortest paths from vertex B are:

***********************
*  Vertex   Distance  *
*     A        80     *
*     B         0     *
*     C        79     *
*     D       157     *
*     E       153     *
*     F       153     *
*     G       220     *
***********************

Main Menu
  1  Input a graph from the keyboard
  2  Input a graph from a file
  3  View graph
  4  Single-source shortest path
  5  All-pairs shortest path
  6  Topological sort
  7  Exit the program

Enter your selection[1-7]: 7

Good bye.
```

The string “`A-B: 80`” means that there is an edge with weight 80 from vertex A to vertex B. The string for a disconnected vertex is simply the vertex label. For directed graphs, use the notation “`A>B`” to indicate a directed path from vertex A to vertex B. You do not need to deal with weighted directed graphs for any of the algorithms.

The output from the all-pairs shortest paths algorithm should be a matrix like the example below (run on Graph 1 above) containing the shortest distances between all pairs of vertices in the graph.

	A	B	C	D	E	F	G
A	0	80	159	237	233	233	300
B	80	0	79	157	153	153	220
C	159	79	0	78	176	74	141
D	237	157	78	0	98	89	156
E	233	153	176	98	0	187	254
F	233	153	74	89	187	0	67
G	300	220	141	156	254	67	0

Extra Credit

We will award between 5-15 points extra credit for turtle graphics representations of graphs and results.

Project Deliverables

Submit your Python source files to the Moodle drop-box by the due date and time.

General Policies & Grading of Projects

Ground rules:

- Start early, and do not hesitate to seek early feedback from me!
- Your final submission should represent ***your*** understanding of the problems.
- You can talk with your classmates about this problem, but do so in terms of general ideas of how to solve, on a whiteboard. While it is OK to use code from your textbook and standard Python libraries, never copy code from another student or the Internet, and never give your code to someone else. This constitutes plagiarism and you will be subject to academic penalty and/or a disciplinary sanction by the University.