CS 340 Programming Assignment V:
Single-Source Shortest Paths

Description: You are to implement the following algorithms: DAG SP algorithm, Dijkstra's algorithm, and the Bellman-Ford algorithm for single-source shortest paths based on (i) whether or not you detect a cycle and (ii) whether or not you detect any negative edge weights. If there is no cycle, then run the DAG SP algorithm. If there is a cycle but no negative edges, then run Dijkstra. Otherwise, run Bellman-Ford.

I/O Specifications: You will read your input graph from an input file named **graphin.txt** of the weighted adjacency list representation and format identical to that of the previous project (although this time the graph is directed and weights can be negative). You need to prompt the user for the source node. Moreover, continue to prompt the user for the destination node, allowing the user to select different destination nodes based on the same source. Regarding your output, which will be to the console:

- First state whether or not the graph is a DAG, and if so indicate that you will run "DAG SP algorithm". Otherwise, state whether or not the non-DAG graph has any negative edge weights. If the graph is not a DAG and has only positive edges, indicate that you will run "Dijkstra's algorithm". Otherwise, state that the graph is not a DAG and has negative weight edges, indicating that you will run "Bellman-Ford algorithm".
- If a negative weight cycle exists, output this fact to the console and exit.
- Otherwise, regardless of the graph type, output both the shortest path from the source to the destination, and the total length of that path.
- Continue to prompt for another destination until the user quits.

Algorithmic Specifications: You need to check the graph for the existence of any cycle in the graph using DFS, and if no cycle is detected (i.e. no back edges) you will run the topological-sort based shortest path algorithm for DAGs taking O(E+V) time. If the graph is not acyclic, but has only positive weight edges, you will run Dijkstra's algorithm which takes O(E log V) time. Otherwise, you will run Bellman-Ford algorithm (taking O(EV) time). If you detect a negative weight cycle in the graph, you will output this fact to the console and exit. Otherwise, you will keep prompting the user to input a destination node whose shortest path from the source you will output to the console, including both the nodes along the shortest path and the overall shortest path cost.

What to Turn in: You must turn in a single zipped file containing your source code, a Makefile if needed, I/O files, and a README file indicating how to execute your program (especially if not written in C++ or Java). If your program cannot be compiled and/or run on home.cs.siue.edu, you must show me your code running in office hours.

This assignment is due by MIDNIGHT on Tuesday, October 31st. Late submissions carry a -40% per day penalty.