CS 323 Homework 5

Due by Sunday, March 19, 2017 8:45 PM

Submission instructions

Submit your assignment through the QTest system, using course ID: **CS323** and exam ID: **hw05**. Write all your code into the code boxes provided in QTest, and make sure that it works correctly by pressing the "Execute" button. If your program is composed of multiple classes, make the first one public and all the other ones not public (just omit the public visibility modifier for all classes except the first one). No email submissions are accepted. No late submissions are accepted. Include a collaboration statement in which you acknowledge any collaboration, help, or resource you used or consulted to complete this assignment. This section must be written even if you worked on the assignment alone.

1 Graph Algorithms (75 points)

Add the following methods to the code you developed in Homework 4.

- **(5 points)** `public double totalWeight()` Returns the sum of the weights of all the edges in this graph.
- **(10 points)** `public Graph prim(String root)` Returns a new Graph representing a minimum spanning tree of the original graph, calculated using Prim's algorithm. You can use your heap implementation from Homework 3.
- **(30 points)** `public Graph kruskal()` Returns a new Graph representing a minimum spanning tree of the original graph, calculated using Kruskal's algorithm. You will also need to implement the necessary operations for a disjoint-set data structure in order to efficiently implement Kruskal's algorithm.
- **(10 points)** `public boolean bellmanFord(String source)` Runs the Bellman-Ford algorithm on the current graph. The method returns `false` if a negative cycle is detected, `true` otherwise. The information necessary to extract the shortest paths gets stored in the current graph, in the same attributes that are also used by DFS and BFS.
- **(10 points)** `public boolean dijkstra(String source)` Runs Dijkstra's algorithm on the current graph. The method returns `false` if the graph contains negative edges, true otherwise. The information necessary to extract the shortest paths gets stored in the current graph, in the same attributes that are also used by DFS and BFS.
- **(10 points)** In the main method of your program, write enough test cases to thoroughly test all the methods you implemented. To test Bellman-Ford and Dijkstra, you will use the `path` and `pathWeight` methods you implemented in Homework 4 (you may need to make some minor modifications to these two methods depending on how you originally implemented them).

2 Emory Airways (25 points)

You are the CEO of a new airline: Emory Airways. The airline will initially connect a network of 5 airports of your choice in the United States.

- **(10 points)** Using the graph library that you developed so far, create a fully connected graph where the vertices represent the airports served by Emory Airways (use IATA airport codes as vertex names). The weights of the edges represent the flight distances between airports. You can lookup IATA codes and flight distances on this website: `https://www.world-airport-codes.com/`
- **(5 points)** Assume that maintaining a direct connection between one airport to another costs an amount of money proportional to the flight distance between the two airports. Using your previous implementation of Prim's algorithm, find how to simplify your network so that it is still possible to reach all the airports but the total cost of the network is minimized. The root parameter of Prim's algorithm is the headquarters of your airline.
- **(10 points)** Using your program, Graphviz, and LaTeX, prepare a PDF report where you show the two graphs representing your network, the total cost of the fully connected network, and the total cost of the simplified network.

Grading criteria

- Programs that do not compile in QTest get zero points.
- -10 points for missing collaboration statement.
- -10 points for missing comments or bad usage of comments.