CSCI 230 -- PA 11

Shortest Paths

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Exercise 1: You should use **AdjacencyListGraph** class from previous PA, which contains node/edge information and **adjacency list using a list or vector**. If you had problems with the class from previous PA, you could use **AdjacencyMapGraph** class. Download the data file, ***PA11Flights.txt***, for this PA. The file contains flight connections between airports (a directed edge) – source airport code, destination airport code, and cost of the flight. Construct the digraph from the data file and display the graph like previous PA.

Exercise 2: Modify exercise 1 to include additional features and you can just submit exercise 2 since it includes all features of exercise 1. Additional graph processing algorithms such as shortest paths can be added to this class or another class such as **GraphAlgorithms**. You should try to utilize existing classes in previous PA (modify as needed) and graph algorithms discussed in book/class if applicable. Provide the user with following menu:

0. Display all flights
1. Find a cheapest flight from one airport to another airport
2. Find a cheapest roundtrip from one airport to another airport
3. Find an order to visit all airports starting from an airport

Q. Exit

Your program will continue until the user wants to exit this program.

For option 0, it displays all vertices and edges in a readable format (exercise 1). If two airports are needed, then you must enter source airport code following by destination airport code. For option 1, user will be prompted for the two airport codes; if the connection is found, display the cheapest flight -- all airport codes starting with the originating airport, cost of that flight, and total cost (like LAX – $189.00 - - > DFW,

$189.00). Option 2 is like option 1, but you need to provide both departing flight and returning flight if a roundtrip is possible. For example, display the cheapest flights -- all airport codes starting with the originating airport and total cost (like LAX -- $189.00 - - > DFW and DFW – $99.99 - -> SFO -- $79.00 - - > LAX, $367.99). For option 4 is like option 1 but select a flight with fewest stops. For option 3, enter a starting airport and provide an order to visit all airports if it is possible; otherwise, indicate not possible.

You must run the following test cases in this order as one set of test cases and feel free to try more test cases as needed:

1. Run option 0
2. Find the cheapest flight from LAX to JFK (option 1)
3. Find the cheapest flight from JFK to LAX (option 1)
4. Find the cheapest roundtrip from LAX to JFK (option 2)
5. Find the cheapest roundtrip from SEA to SFO (option 2)
6. Find an order to visit all airports starting from LAX (option 4)
7. Run option Q

Question 1: What is the running time for your shortest path algorithm in big-O? Explain.

Question 2: Should you just run all-pairs shortest once and use the results for options 1 and 2? Explain why or why not.

Extra Credit: Add option 4 to exercise 2 to find a flight with fewest stops from one airport to another airport. You can just submit this final version since it includes all features of exercises 1 and 2. Run the following cases: find a flight with fewest stops from JFK to LAX and find a flight with fewest stops from SFO to SEA.

Fill out and turn in the PA submission file for this assignment (save as PDF format).