

## CS241 Spring 2018 Homework 4

Total points: 100

Due date: Friday, June 1st, 2018 at 11:59pm

### Purpose:

Implement several methods of a graph data structure and use it in a practical application.

### Task Description:

Input: There are two files to be used with your project - one contains city data and the other contains road data.

- **city.dat:** This file contains information about cities, where each line has 5 attributes: *City Number*, *City\_Code (2 letters)*, *Full\_City\_Name*, *Population*, and *Elevation*.
- **road.dat:** This file contains information about roads, where each line has 3 attributes: *From\_City*, *To\_City*, and *Distance*. Note that all roads are assumed to be one-way. *From\_City* and *To\_City* contain the *City Numbers* of their respective cities.

You will need to update the Vertex class to keep all the required information for each city, i.e. the population, elevation, *City\_Code*, and *Full\_City\_Name*. In the graph's HashMap, the city number is the key and the vertex is the value.

When the program is run, it should display a menu-driven system with the following options.

- Read the original data files and store the data to appropriate data structures.
- Allow the user enter a *City Code*. Your program should print out the city information (the whole record).
- Find the connection between two cities.
  - The user will be asked to enter two *City Codes*. The program finds the **cheapest** distance between the two cities. Note: You will need to implement the method `getCheapestPath` in `DirectedGraph.java`, which should take into account the weights on each edge. Do not use the already implemented `getShortestPath`.
- Insert a road (edge) between two cities
  - The user will be asked to enter two *City Codes* and a *Distance*. Note that if an edge already exists between a pair of *City Codes* or if one of the *City Codes* doesn't exist, a warning message should print.
- Delete a road (edge)
  - The user will be asked to enter two *City Codes* for a road. Note that if the road entered doesn't exist, print out a warning message. You will need to implement the `remove` method in `NeighborIterator`

- Exit.

Your program should resemble the following output (the user inputs are underlined):

Command?

Q Display city information by entering a city code.

D Find the minimum distance between two cities.

I Insert a road by entering two city codes and distance.

R Remove an existing road by entering two city codes.

E Exit.

Command? Q

City code: LV

12 LV LEE VINING 8390 5983

Command? D

City codes: CH PM

The minimum distance between CHINO HILLS and POMONA is 143 through the route: CH, xxx, ..., xxx, PM.

Command? I

City codes and distance: GG BO 100

You have inserted a road from GARDEN GRPVE to BOSSTOWN with a distance of 100.

Command? R

City codes: KV MP

The road from KERNVILLE and MOUNTAIN PASS doesn't exist.

Command? E

### **Programming Guides**

You may use the code that is on Blackboard under Code->In-Class Exercises- Graphs. The methods you need to implement already exists in stubs. You will need to change the driver program considerably.

### **What to Submit?**

1. A brief report, detailing any trouble you had with the implementation and what, if anything, you learned from this assignment.
2. Source code of any changed files (i.e. DirectedGraph.java and Main.java)

You will be graded based on the quality of your program, your project report, and how well you follow the project description.