|  |  |
| --- | --- |
| Computer Science 211 Data Structures and AlgorithmsFall 2016 |  |

**Lab Report – City Project on Graph Theory**

*Corey Acri*

**Assignment Analysis and Design**

*In your own words, describe the problem including input and output. Briefly describe how you developed your code. Briefly describe what your code does and how it works – including anything different or unique or special that you did in your software. If the software is long or complicated, describe how it is organized.*

***You must create and include*** *a copy of either pseudocode or a UML diagram that you created for this programming assignment. You can paste the pseudocode or UML diagram into this section of your lab report.*

*If you worked with anyone else, asked anyone for help, or looked anything up, then mention it here. Include proper references to source material.*

This project required us to implement Dijkstra’s Algortithm to find the shortest path between two cities. The program asks the user for the origin city and the destination city, prints the city by city steps from the two and then also draws the route on the map.

I worked off of Professor Herbet’s base code which created the Vertex, Edge and City arrays, as well as the classes and methods to read the data for the program and print it to the map.

I added several methods, the primary being dijkstratize(), which runs Dijkstra’s algorithm and it’s corresponding helper methods.

I also added a “CityStack” class for the final step of the algorithm and modified the CityMap class into RouteMap to print only the point to point route at the end.

This was a helpful reference: https://www.cs.usfca.edu/~galles/visualization/Dijkstra.html

**Assignment Code**

*Include the code for your assignment. Unless otherwise directed by the assignment or by your instructor, this will be a zipped copy of your NetBeans project to be uploaded to Canvas as a separate file.*

*This program is quite large, please see attached Netbeans Project for all pseudocode.*

**Assignment Testing**

*Describe how you tested this program to verify that it runs correctly.*

Most of the iteration of the program and testing was done by printing output to the console to ensure that I was properly capturing Dijkstra’s method.

This way of testing allowed me to discover **one major bug** in the code. When you set the default best value in the City object to Integer.MAX\_VALUE and perform a math calculation on it, it results in a negative value. Therefore, I reset the max values to 999999, knowing that we would never have a best distance higher than that based on the values in the distance tables.

I also tested it on several users asking them to print start and end points. I added “list cities” as an option so users can see the proper format for the city name entry and also ignored the case for comparing those strings to the strings stored in the city objects.

**Assignment Evaluation**

*Briefly describe what you learned from this project. What things did you struggle with? What was easy? Give your opinions of the process, including what you liked about the project and any suggestions you have for improving the project.*

This was the most rewarding project to date since we did Monopoly and the password checker code in 111.

Implementing the algorithm forced me to really learn and understand it. Plus, this program turns out to be an absolutely excellent tool for cross-country trip planning!!!

Finally, graph theory, as we have discussed, is highly relevant to my efforts to build a company that provides personalized bike directions to riders to enhance riders’ perception of safety.

Thank you!