

/******

* Project Report Template

* Project 3 (Map Routing), ECE368

*****/

Name: Matthew Bolda

Login: MBolda

/******

* Explain your overall approach to the problem and a short

* general summary of your solution and code.

*****/

I started at the first file, made a helper function to read the numbers one by one. This helped me get the vertices into a list and create an array for them. Then I would create an array of the connected vertices. From there I moved to the second file, the first would be the source and the second would be the destination. I would put the source into the queue and make the distance 0, then I would put all its neighbors in the queue and go until the queue is empty, updating the distance and such. Once the distances are done I would go to the destination vertex and print the previous vertex until I reached the source. If the distance is still infinite, then the vertices are not connected and you should print "INF". Continue this process for every query.

NOTE: I used the math library for the pow function. I used the pow function to find the distance between two vertices.

/******

* Known bugs / limitations of your program / assumptions made.

*****/

My program does not always get the shortest path but sometimes gets a path that is very close to the shortest. This is because I failed to completely finalize the distance for some nodes.

/******

* List whatever help (if any) that you received.

*****/

Even though we learned about queues in class I used geeks for geeks to learn more/brush up and this helped guide my functions that corresponded to using Queues. I only used this source for creating my queue and the simple operations on the queue.

LINK: <https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/>

A youtube video I used for learning more about Dijkstra's Algorithm helped me create my strategy for this project. The video had no code or anything other than visualizations to help guide me.

LINK: <https://www.youtube.com/watch?v=pVfj6mxhdMw>

/*****

* Describe any serious problems you encountered.

*****/

Before I had a queue it would not visit all nodes. I have had a very difficult time trying to change the queue to a priority queue.

/*****

* List any other comments/feedback here (e.g., whether you

* enjoyed doing the exercise, it was too easy/tough, etc.).

*****/

Something to note about my program is that before it outputs any paths it takes a long time to initialize all the information needed. Once the information is initialized finding the shortest path for any query is very quick.

The compile line I used for this project was:

gcc -Werror -Wall adjacent.c -o adjacent -lm

so that

./adjacent (insert map.txt) (insert query.txt) will start the program.

This is because I used the math.h library and could not get it to work without the -lm flag.