

/******

*** Project Report Template**

*** Project 3 (Map Routing), ECE368**

*****/

Name: Abhimanyu Agarwal

Login: agarw184

/******

***Explain your overall approach to the problem and a short**

***general summary of your solution and code.**

*****/

Implementation:

Min Heap, data-structure that would be used to store each vertex of the map.

The following approach could be adopted in order to successfully implement the algorithm:

Steps:

- 1) Create an array that would store the vertices stored from the (first input file) map.
- 2) Build an adjacency list using the data from the map.
- 3) Check the number of queries by reading the query file.
- 4) We check the queries until we hit the end of file. It is done one by one.
- 5) Address of each vertex is stored in a different array.
- 6) We then build a min heap with all the vertices of the map.
- 7) We swap the first and the visited node in min heap and do down heap traversal.
- 8) Perform up-heap traversal after each element in the adjacency list of the elements swapped in updated.
- 9) We loop through this entire process until we encounter the heap to be empty or if the destination element has been marked by us.
- 10) Move on to the next query and do the same process for it as well.

Output:

Using the above steps for implementation, we can print the shortest path from the source vertex to destination vertex along with its distance.

Files submitted:

shortestpath.c

header.h

report.pdf

/******

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

*****/

While compiling, it is requested to include '-lm' flag due to inclusion of the in-built math library.

Compilation command: **gcc -g -Werror -Wall -lm <filename.c> -o <filename>**

/******

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

*****/

Referred to Wikipedia in-order to understand the algorithm.

Used geeks for geeks and code academy for the pseudo-code on using the algorithm and making use of the min heap in the implementation.

/******

***Describe any serious problems you encountered.**

*****/

No major problem encountered as such.

/******

***List any other comments/feedback here (e.g., whether you enjoyed doing the exercise, it was too easy/tough, etc.).**

*****/

Overall, I feel this project was hard. It did require a lot of thinking. I had to first read about the algorithm and other algorithms that were associated or like Dijkstra's algorithm. Therefore, I found the project challenging. To conclude, I enjoyed the course and new things that I learnt throughout the semester. Moreover, I learnt how useful the heap could be by applying it in this project