

```

/*****
 * Project 3 (Map Routing) Final Report, ECE368
 *****/

```

Name: Kunwar Digraj Singh Jain

Login: jain126

```

/*****
 * Explain your overall approach to the problem and a short
 * general summary of your solution and code.
 *****/

```

COMPILE INSTRUCTION: gcc -Werror -Wall -O3 dijkstra.c -lm -o dijkstra

A graph data structure is used to make an adjacency list which contains a node and the nodes surrounding it (adjacent, connected with an edge). Then the source is given a value of 0 as its weighted distance and all the other vertices are given a value of -1 for their distance weights. Once this is completed, we start with the source node and get the tentative distance to all the surrounding vertices based on the adjacency list. The source node is then added to the visited vertices array. We then call a function which gets us the next vertex that should be taken into consideration based on the smallest tentative distance. Once we receive the index of that vertices, tentative distance to all its surrounding nodes is calculated (these only include nodes which are not present in the visited array). This vertex is then added to the visited array. This process is continued till the destination vertex is not added to the visited set. Once this step is reached we have the shortest distance. By the time we obtain our shortest distance, another array called previous stores all the vertices which were taken to obtain the shortest distance. We then print the shortest distance and the path that was taken to obtain it.

```

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

```

As of now, there are no known bugs.

A limitation of the program is that if the query file has destination and source vertices which are very far away, it will take a long time to compute the shortest distance. If there are only a few of those, it will take around a minute to compute the shortest distance and the path taken to obtain it.

Some of the assumptions I made were that the highest number of vertices would be 100000 and the coordinates (x or y) would be smaller than 10000. I also made an assumption that all the distance calculated would be whole numbers (positive integers).

```

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

```

I received help regarding the next vertex calculation from the professor. He helped me figure out that I could use an array to find the shortest tentative distance and thus calculate the vertex that should be used next.

```

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

```

One problem that I faced was that when two cities were placed on the same index, the program segment faulted. But it was fixed later.

Another problem was the recreation of the path since I did not use a min-heap.

```

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

```

I really enjoyed this exercise. It was my favourite project of all three. It was neither too tough nor too easy. It was a good project that tested a lot of things that I learnt in this course.