```
/*****************************************************************
 * Project Report Template
 * Project 3 (Map Routing), ECE368
 *****************************************************************/

Name: Chul Woo Park
Login: park724

/*****************************************************************
 *  Explain your overall approach to the problem and a short
 *  general summary of your solution and code.
 *****************************************************************/
```
In this project, I used Dijkstra algorithm in order to find the shortest distance and path. I used three forms of data structure. Firstly, I have a MapNode which contains information such as xpos, ypos, distance from starting node, visited flag and adjacency linked list. Distance is initialized to infinity in the beginning of my program. Secondly, I have a AdjNode, which is the node for the adjacency linked list. Lastly, I have a PathNode, which is the node for the path linked list. I implemented such algorithm by using the recursive function. I have two base cases for my recursive function, which are return if distance computed so far is greater than the previous shortest distance and return if current position is the ending point. Basically, if it enters my recursive function, it means that such node passed as parameter is visited. Then, I go through the entire adjacency linked list and compute the distance from that node. If it's less than the previous distance, I update it with the new distance, otherwise, it's just ignored. Then, I go through the entire node in map and find the node that has the shortest distance from the starting note (from query). Then, I pass that node to the recursive function call, which means that it's going to visit that node next. After all of these recursive function calls, distance variable in the node of map will contain the shortest distance since it has been updated every time if the shorter distance was found.

```
/*****************************************************************
 *  Known bugs / limitations of your program / assumptions made.
 *****************************************************************/
```
Known bug in my program is that it doesn't print the shortest path, while it still prints the shortest distance correctly. I originally had the path being printed, but I took it off because it took so much time for large input files, such as usa.txt. The reason that printing the path was problem is explained below. In addition, my program is quite slow and the timing report is also written below.

```
/*****************************************************************
 *  List whatever help (if any) that you received.
 *****************************************************************/
```
No help was received.

```
/*****************************************************************
 *  Describe any serious problems you encountered.
 *****************************************************************/
```
My original approach to this project was very similar to the maze algorithm. Basically, the algorithm was going through all possible paths and updates the distance and path if shorter one is detected. In such way, at the end of my program both the shortest distance and shortest path were correctly found. However, the problem was that it took so much time for large input files. I think it was because I was going through all possible paths and those are way too many. I couldn't cut down some possible paths that are not going to be the shortest path. Therefore, I decided to change my algorithm that kind of implements the priority queue. I did that by finding the node that has the minimum distance from the starting node.

However, it didn't work as fast as I thought because I had to go through the entire map to find that node. It could have been done by using the min_heapify, but I couldn't think of that when I was coding this project.
usa1.txt: It took 0.290000 seconds for finding shortest path(s)
usa10.txt: It took 86.339996 seconds for finding shortest path(s)
usa100.txt: It took 112.279999 seconds for finding shortest path(s)

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

This project was one of the most challenging assignments I've ever had. In addition, it's my first time that I couldn't make the assignment in a perfect condition. I think that the biggest problem was that my first approach was like doing the maze assignment and Dijkstra algorithm was quite different from that. However, I was able to get the correct shortest distance for all the input files, even large ones. Therefore, I'm fairly satisfied for what I've done. The biggest problem was that I was not quite understanding the algorithm at the beginning and I was in a wrong track. Next time, I should have a better understanding and think thoroughly before starting the code.