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

Name: Kshitij Jain
Login: jain98

/***
 * Explain your overall approach to the problem and a short
 * general summary of your solution and code.
 ***/
I started this problem with a vague understanding of the application of Dijkstra's algorithm because of which I ended up implementing a wrong solution which was not implementing Dijkstra's algorithm correctly. I then read up more on the algorithm online and then finally understood what was going on after which I was able to understand that in each pass, the closest node to the map and not the closest node to the most recently included node, is included in the map.

/***
 * Known bugs / limitations of your program / assumptions made.
 ***/
The program assumes that the input size would not be more than 100000 vertices. If there are more than 100000 vertices, the stack and the heap might clash as there would be not enough space.

The program is also comparatively slower than the sample program given to us for testing. The reason is that I am not using a min heap for extracting the closest element, that I have not previously visited.

/***
 * List whatever help (if any) that you received.
 ***/
I was first creating an adjacency matrix because of which I was filling up the stack and getting memory errors. I then created an adjacency list on the heap, along with the vertices array on the stack. Raiyan helped me figure this out.

/***
 * Describe any serious problems you encountered.
 ***/
I spent one whole day trying to implement a min heap on an array of vertices, but was unable to do so. I also tried creating a list of vertices and then implementing a min heap, but that also did not work.

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

I enjoyed doing the assignment, but I am still curious as to how we would have implemented a min heap on an array of vertices.

Note- My program needs to be compiled with the '-lm' flag