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

Name:       Nathan Moore
Login:      moore669@purdue.edu

Compile and link with -lm.
/*****************************************************************
 *  Explain your overall approach to the problem and a short
 *  general summary of your solution and code.
 *****************************************************************/
```
In my solution, I expand outward from the starting point and add each node adjacent to my priority queue. To do this,I store the matrix in an adjacency list since it supports my required actions faster then an adjacency matrix. My priority queue is implemented using a binary heap.

My code logic is as follows. I first read in the graph. Since I store the data as an adjacency list, I first initialize every array that stores the index of the adjacent node to be the number of edges over vertexes (assuming the edge distribution to be about constant) with an extra buffer space for variability. I resize these arrays if necessary. I read the queries in one at a time when necessary. When I search for the shortest path, I take efforts to reduce the calls to upheapify by replacing the node with a node that I'm inserting if possible. Otherwise I add the node to the queue if it hasn't been found yet and remake the heap.

```
/*****************************************************************
 *  Known bugs / limitations of your program / assumptions made.
 *****************************************************************/
```
I assumed that the edge weight would be less then or equal to 10k. I also assumed that the number of queries could fit into a 16 bit number.
There are no known bugs in my program, but for graphs with a very large number of edges, the program's performance would be terrible due to having to resize the heap.

```
/*****************************************************************
 *  List whatever help (if any) that you received.
 *****************************************************************/
```
I got some implementation ideas from CLRS.

```
/*****************************************************************
 *  Describe any serious problems you encountered.
 *****************************************************************/
```
The hardest part was getting my program to fit into memory and testing the correctness of my program. The later was worked out by writing a program to walk along the paths given by my program and the sample binary and comparing the integer lengths.

```
/*****************************************************************
 *  List any other comments/feedback here (e.g., whether you
 *  enjoyed doing the exercise, it was too easy/tough, etc.).
 *****************************************************************/
```
I thought this assignment was fairy easy. I also fairly enjoyed it, but that is likely because I ended up automating almost everything with a makefile.