

```

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

```

Name: Nadav Zinger

Login: nzinger

```

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

```

My Overall approach was to use the Dijkstra's algorithm given in the prompt from Wikipedia. In order to execute this algorithm correctly, I had to make several different data structures. The first data structure I used was a 2-d array that stored the vertex in the first row, its x coordinate in the second row and the y coordinate in the third row. The matrix was a 3 by # of vertexes matrix. This data structure was used to calculate distances between two adjacent vertexes. The second data structure I used was an array of linked lists where the index contains the first struct containing the vertex and a pointer to the next struct. The next struct contains an adjacent vertex to the original indexed struct. When there are no more adjacent vertexes the next pointer points to null. The third data structure was an array of structs that was used as my queue. Each struct contained the vertex number and its tentative distance. An array called visited was also used for me to help determine which vertex I had the minimum distance that needed to be popped (u). I also had an array that stored tentative distances at the index called dist and an array that kept track of the path called prev.

```

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

```

My program has no bugs/limitations however since I do not use a priority queue/minheap my program might not run as fast as others. Also a couple test (1-2) cases have the wrong path (according to shortest path executable).

```

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

```

Wikipedia and student collaboration

```

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

```

I experienced a serious problem where my map5 and query5 worked but my U.S was completely wrong. I realized my adjacency List was not correct (forgot to link the second value to the first, only linked first to second must link both).

```

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

```

I enjoyed this exercise, the satisfaction of completing it felt awesome. The exercise was tough but I thought project 2 was the most challenging prompt thus far.

To compile: gcc -Werror -Wall shortest_path.c -o shortest_path -lm

To run: `./shortest_path file1.txt file2.txt`