

Project 3 (Map Routing), ECE368

Name: Ethan Glaser

Login: glasere

Files: map.c, map.h

Compilation: compiled with the math library, add -lm to the compilation command

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

My approach to the problem varied as my understanding of the goals of the project increased. At first I tried implementing some functions to determine all possible paths from point A to point B and identify the least distance option. Then I tried taking the most direct path, where the next node was selected based on the difference between the absolute distance of the origin and destination to the origin to the point to the destination. Neither of these worked well and I then looked further into Dijkstra's algorithm, which I implemented and successfully ran. My initial solution had a 2D square matrix with each point present in the x and y axes, and the distance between coordinates represented by the array crossover point (array[x][y] and array[y][x]). This approach led to a successful solution, but very slow and extremely heavy memory usage in some of the more complex cases. I then reduced the memory consumption by using an array of list nodes that pointed to adjacent values instead of an entire row of a matrix dedicated to each point. This drastically reduced memory consumption and also improved the speed of the algorithm.

Known bugs / limitations of your program / assumptions made.

Compiled with the -lm (math library). The algorithm is still not as fast as it could be and uses a lot of memory, both of these things could potentially be reduced in a few ways. I assumed that the text files would have coordinates in order and the distances would be integer values. The algorithm may be somewhat limited in larger cases due to the speed.

List whatever help (if any) that you received.

My main help on this project was gained by watching some videos on Dijkstra's algorithm after I realized my own solution implementation wasn't feasible. I talked to my friends to get some insight as to why my initial program was running slow and that is what inspired me to use an array of link lists.

Describe any serious problems you encountered.

As I mentioned previously, I initially struggled with my approach to the problem. Once I figured that out, timing and memory usage were major issues.

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

A very good project that relates to what we are learning in class, has real world applications that are relevant, and fun to do as well. It wasn't too difficult once I had an understanding of how Dijkstra's algorithm worked. The projects in this class have provided interesting insight on how real-world problems are solved (compression, mapping).