ECE 368 Project 3 Report

Xiao Xiao
0026557235

Overall

The main purpose of this project is to implement Dijkstra's algorithm to find the shortest path between two points in a map represented by a graph. Two input files are given to provide the connections between nodes and desired starting points and destination.

Summary of the solution

The major approach of my project is to use linked list to implement a graph which contains all the connections between each point in the map. First step is to load the data from the file into the graph. The head node of each linked list in the graph will store the x and y coordinated, which will be used to calculate weights of each edge. A linked list contains all the vertices connected to the node is followed by the head node. Then a min heap is built to implement a priority queue which will place the vertex with shortest distance from the source on top of the min heap. Two arrays are created to store the distance and previous node number. Each time a vertex is extracted from the min heap, it will be marked visited and all the vertices connected to the node will have their distances from source updated. After all the elements inside the min heap are extracted, the values of distance and previous vertices stored in arrays will be used to printout the answer.

Assumptions and Limitations

All the input files will follow the same format as the given sample files.
There will not be any negative values for edges.

Problems Encountered

At the beginning of the project, the construction of the graph cost me a lot of time. Then the min heap part was also comlicated. The difference between decrease key operation and upward heapify made me struggle for one day. But after the min heap was built, the process went smoothly. Then I found a significant bug in my project. Some of the nodes only have single direction connections, which is absolutely not what I want. Then I changed the load file function to make each node inside corresponding linked list of a head node will be assign the head node as one of the nodes inside their linked lists.

Feedback of the Project

This project is no that long if the major idea of the Dijkstra's algorithm is understood. And not like project 2, once the outputs for small samples are correct, the outputs for large size inputs will also be handled. The difficulty of this project is much better than the last one.

math.h library has been included, lm flag need to be added when complie.