

Peter Sumner

### ECE368 Project 3 Milestone 1

The first part of my solution is going to be reading the input file and storing the data. At the beginning of the input file is a line that says how many points and edges are present followed by the points and then the edges. The first line serves as an indicator to know how many nodes and edges to read from the file. To store the data points, I will create a heap where each index corresponds to the point number. The heap will contain a struct object for each point that contains its x and y coordinates and a list of points that are connected through edges. This will also calculate and store the length of each edge. To calculate the distance for each edge, the Euclidean distance will be used.

Once the map data is stored, the algorithm can begin to read the second file. For each pair of nodes in the second file, the distance is initialized to INF. Then the code will go through the heap, starting at the first node and finding all instances of a path that leads to the end node. While doing this, if a path exceeds the length of another completed path, the algorithm will stop on that route. This will help to optimize the time efficiency of the algorithm. If no path is found, the value will not change. The distance of each edge is stored in the heap and the total length will update as the code travels between nodes. To store the order of nodes that are passed, a linked list will be created. After the shortest path is found, the algorithm will print out the shortest distance and then the path took to get this distance. If there are two paths with the same distance, the algorithm will only output the first path found.