

Project 3 Milestone – 1

Abhimanyu Agarwal
agarw184

Objective: Given a map file and a query file as inputs, goal is to compute the shortest path from each source listed in the query file to the corresponding destination using Dijkstra's algorithm. The program takes the name of the map file and the name of the query file.

Input Interpretation:

We would be given 2 inputs files which provide information that would help to implement the algorithm.

- The first input file will represent a map, which is an **undirected** graph whose vertices are points on a plane and are connected by edges whose weights are Euclidean distances.
- The second input file contains a list of search queries with the first line containing the number of such queries and each of the following lines containing one query in the form of a source vertex and destination vertex pair.

Using the information from the first input file, we can estimate the Euclidean distance between every node that is connected. Furthermore, from the second input file, we could calculate the shortest path for each of the queries.

Implementation:

Min Heap, data-structure that would be used to store each vertex of the map.

The following approach could be adopted in order to successfully implement the algorithm:

Steps:

- 1) Create an array that would store the vertices stored from the (first input file) map.
- 2) Build an adjacency list using the data from the map.
- 3) Check the number of queries by reading the query file.
- 4) We check the queries until we hit the end of file. It is done one by one.
- 5) Address of each vertex is stored in a different array.
- 6) We then build a min heap with all the vertices of the map.
- 7) We swap the first and the visited node in min heap and do down heap traversal.
- 8) Perform up-heap traversal after each element in the adjacency list of the elements swapped in updated.
- 9) We loop through this entire process until we encounter the heap to be empty or if the destination element has been marked by us.
- 10) Move on to the next query and do the same process for it as well.

Compilation and Flags:

- `gcc -g -Werror -Wall <filename.c> -o <object_filename>`

Output:

Using the above steps for implementation, we can print the shortest path from the source vertex to destination vertex along with its distance.