horizontal line

|  |  |
| --- | --- |
| **Team IDS** | **Team Members Names** |
| 2016170383 | محمد هشام عثمان أبو المجد |
| 2016170377 | محمد مصطفى قطب حسن |
| 2016170368 | محمد على محمد أبو الفضل |
| 2016170380 | محمد مقبل محمد مصطفى |

Map Routing

**15th December 2018**

# OVERVIEW

In recent days, transportation applications to move from one place to another has been evolving rapidly. The use of data from many users helps to make transportation better. Especially in reducing transportation time and avoid heavy traffic roads.

In this project we provide two versions of Dijkstra Algorithm(Normal Dijkstra, Bi-directional Dijkstra) to find the least time to move from the source location to the destination location, and find the path that achieves that time.

Also we provide a checker to compare the expected output to our output, and get the difference if exists.

Map visualization is provided with the optimal path to the sample cases

# GOALS

1. Learn how to speed up Dijkstra algorithm and use it in the project efficiently.
2. Optimize the constant factor of the overall complexity of the algorithm to speed the running time using the Bi-directional Dijkstra.

# SPECIFICATIONS

* The moving person walks with constant speed (5 km/h) in straight lines (from the source to the starting intersection and from the finishing intersection to the destination).
* The moving person cannot walk more than R meters to move from the source location to the starting intersection. They also cannot walk more that R meters to move from the finishing intersection to the destination location.
* The time to get in a vehicle or get out of a vehicle is negligible (is not taken into consideration).
* The moving person will ride only one vehicle. They cannot get out of the vehicle unless they reached the final node.
* The vehicle always moves with the road speed. It changes speed only if the road is changed.
* The roads connect intersection in straight lines. They are also bidirectional roads.

# BOUNS

## Map visualization

## Bi-directional Dijkstra

# MILESTONES

## Milestone 1

1. Construct a weighted graph for the map (using adjacency list as the map is connected as sparse graph in ϴ(E)).
2. The search for the optimal path to move from the source location to the destination location (small and medium cases both done by normal Dijkstra algorithm O(E log(V)), and Bi-Directional Dijkstra algorithm O(E log(V)).
3. Each query output should contain 5 lines as the following:

* The shortest time to move from the source location to the destination location (in minutes)
* The total distance of the path with the shortest time (in kilometers)
* The total walking distance (in kilometers)
* The total vehicle distance (in kilometers)
* The total execution time (in milliseconds)

## Milestone 2

1. Building the optimal path (output the nodes in order).
2. The project should work in all cases (small, medium, and large cases).
3. Documentation II

# CODE SNIPPETS

# PROJECT RUNTIME SNIPS