Google Map and Shortest Path Dijkstra Algorithm Step By Step

Ahmad Naser

Published with GitBook

Table of Contents

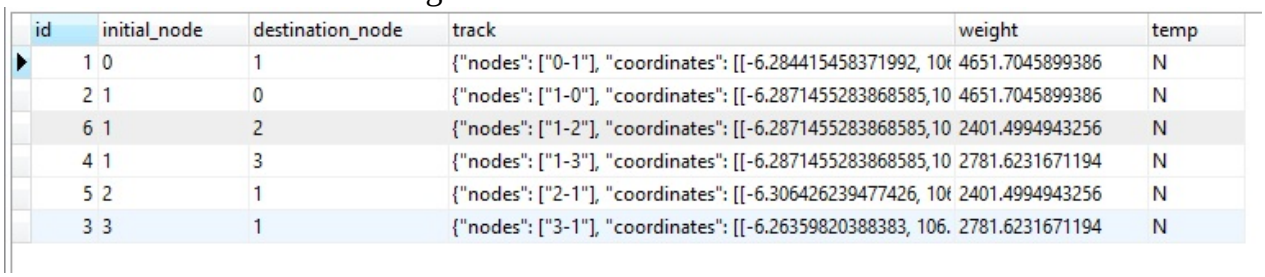
My Awesome Book

Create Graph Array

First we need to have the nodes with their coordinates, and we need to sort them ascending based on their order, we use this query to sort them:

```
SELECT * FROM graph order by CONVERT(initial_node, SIGNED INTEGER),
CONVERT(destination_node, SIGNED INTEGER) asc
```

this sort will result as following:



id	initial_node	destination_node	track	weight	temp
1	0	1	{"nodes": ["0-1"], "coordinates": [[-6.284415458371992, 10(	4651.7045899386	N
2	1	0	{"nodes": ["1-0"], "coordinates": [[-6.2871455283868585,10	4651.7045899386	N
6	1	2	{"nodes": ["1-2"], "coordinates": [[-6.2871455283868585,10	2401.4994943256	N
4	1	3	{"nodes": ["1-3"], "coordinates": [[-6.2871455283868585,10	2781.6231671194	N
5	2	1	{"nodes": ["2-1"], "coordinates": [[-6.306426239477426, 10(	2401.4994943256	N
3	3	1	{"nodes": ["3-1"], "coordinates": [[-6.26359820388383, 106.	2781.6231671194	N

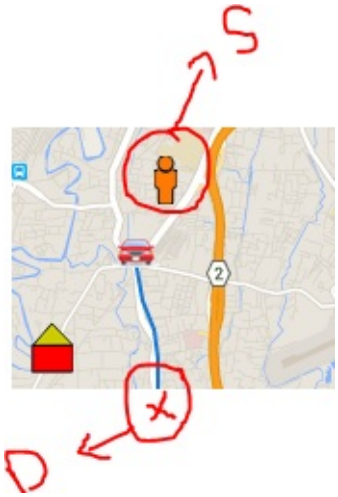
we need to convert this table into a memory table using two dimensional array:

```
* ================================================
* Convert graph into two dimensional array
* ================================================
*
* DIJKSTRA ALGORITHM USED ARE IN DB TABLE, LIKE THE EXAMPLE BELOW :
   --------- ---------- --------- --------- ---------
  |   0     |    1     |    2    |    3    |    4    |
   --------- ---------- --------- --------- ---------
  | 0->1=10 | 1->0=10  | 2->0=11 | 3->4=89 | 4->0=90 |
  | 0->2=11 | 1->2=55  | 2->1=55 |           | 4->3=89 |
  | 0->3=40 | 1->4=20  | 2->3=54 |         |         |
   --------- ---------- --------- --------- ---------
* BEING CONVERTED FROM ABOVE TABLE To 2 Dimensional Array:
  [0][0] = '=> 1->10';
  [0][1] = '=> 2->11';
  [0][2] = '=> 3->40';

  [1][0] = '=> 0->10';
  [1][1] = '=> 2->55';
  [1][2] = '=> 4->20';
```

```
   [2][0] = '=> 0->11';
   [2][1] = '=> 1->55';
   [2][3] = '=> 1->54';

   [3][0] = '=> 4->89';
   [3][0] = '=> 4->89';

   [4][0] = '=> 0->90';
   [4][3] = '=> 3->89';
*/
```

Getting the source and destination

```
koord_user:{"lat": -6.257967164413908, "lng": 106.86830520629883}

koord_destination:{"lat": -6.272983134659221, "lng": 106.858177185
```



we need to get the source (user) latitude and longitude and the destination (target) latitude and longitude in order to calculate the shortest path using the Dijkstra Algorithm.

First Chapter

GitBook allows you to organize your book into chapters, each chapter is stored in a separate file like this one.