In class Lab 12 – SSSP

Name: Meddepola M.A.C.T

Index No: 210383L

GitHub Repository:
https://github.com/Thejas0604/CSE-labs

(01) Weighted adjacency matrix representation

$$\begin{bmatrix} 0 & 10 & 0 & 0 & 15 & 5 \\ 10 & 0 & 10 & 30 & 0 & 0 \\ 0 & 10 & 0 & 12 & 5 & 0 \\ 0 & 30 & 12 & 0 & 0 & 20 \\ 15 & 0 & 5 & 0 & 0 & 0 \\ 5 & 0 & 0 & 20 & 0 & 0 \end{bmatrix}$$

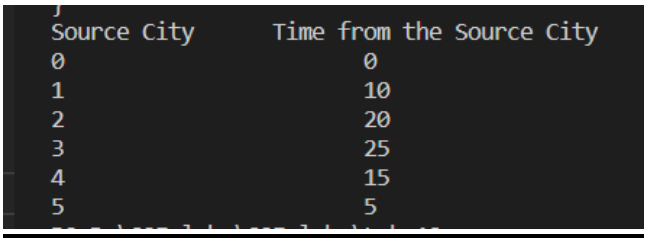
(02)

- We need to place hospital such that the ambulance in that hospital can attend to all cities by the shortest time.
- In our graph cities can be represented using vertices, roads can be represented using edges and time in between cities can be represented using weights.
- Since no negative weights are being involved, we can easily use Dijkstra's Algorithm for this scenario.
- Check out my implementation on this scenario using Dijkstra's Algorithm in https://github.com/Thejas0604/CSE-labs under Lab 12.

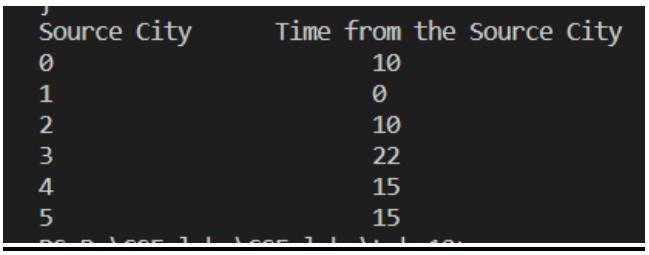
- This code implementation was referred from https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

(03)

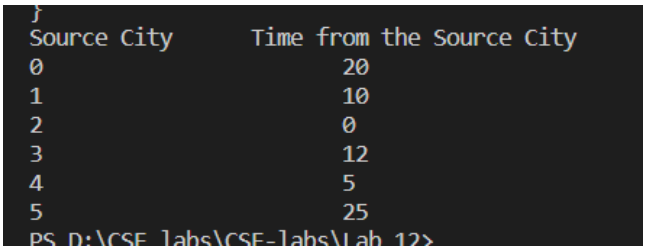
0 as the source



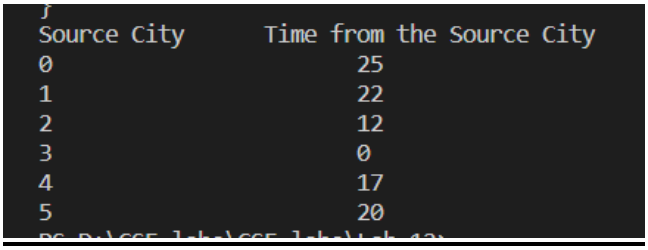
1 as the source



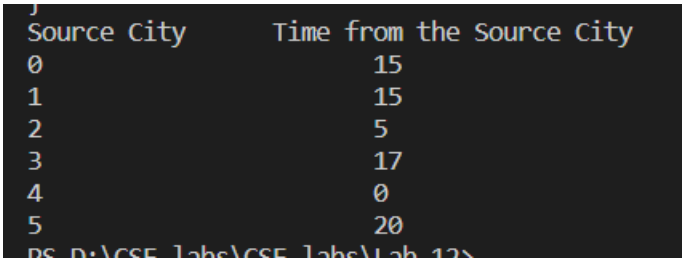
2 as the source



3 as the source



4 as the source



5 as the source

```
}
Source City      Time from the Source City
0                       5
1                       15
2                       25
3                       20
4                       20
5                       0
PS D:\CSE labs\CSE labs\Lab 12>
```

(05)

- Let's add these values to a table and find the averages among them.

Source	Other Cities	Time	Average Time
0	0	0	15
	1	10	
	2	20	
	3	25	
	4	15	
	5	5	
1	0	10	14.4
	1	0	
	2	10	
	3	22	
	4	15	
	5	15	
2	0	20	14.4
	1	10	
	2	0	
	3	12	
	4	5	
	5	25	
3	0	25	19.2
	1	22	
	2	12	
	3	0	
	4	17	
	5	20	
4	0	15	14.4
	1	15	
	2	5	
	3	17	
	4	0	
	5	20	
5	0	5	17
	1	15	
	2	25	
	3	20	
	4	20	
	5	0	

- Since there are 3 cities with an average time of 14.4 (Time units) we can either choose city 01 or city 02 or city 4 as the cities which can place a hospital.