****

**School of Computer Science and Engineering**

**"SHORTEST PATH BETWEEN TWO CITIES BASED ON DISTANCE AND TOLL AMOUNT BY RUNNING DIJKSTRAS PARALLELY WITH TOLL AMOUNT AND DISTANCE"**

# SATYAM CHOUKSEY 15BCE0838

VIJAYKUMAR K

Assistant Professor

**Submitted by**

Certificate

This is to certify that the project work entitled “shortest path between two cities based on distance and toll amount by running Dijkstras algorithm parallely with distance and toll amount ” that is being submitted by “ satyam chouksey” for operating system (CSE 2005) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source.

Place : Vellore

## Acknowledgement

We have done this project under the guidance of professor Vijaykumar k. I heartly thank him on discussing this project with me. He cleared my doubt in a short full way.

## Abstract

I have seen many maps on internet like google map after using that the one thing that comes to my mind is that how it actually works. for example when ever we need to go from one place to other we just write our source and destination and it shows shortest path between those two places that's why I have taken dijkstras to solve this and I have successfully done that.

Project Description

Device a procedure for finding the shortest path between a pair of cities based on the collective information about the distance & the toll amount in the path implement it with a sample code:

First we have taken out the shortest path by using dijkstras algorithm and that finally whatever nodes we travelled in map i provided same link to another graph having same cities but differently weighted (toll amount) now whatever shortest path has take out from dijkstras algorithm it will also travel same path and will carry out the sum of toll this is how we can get shortest path(from source to destination) and corresponding toll amount.

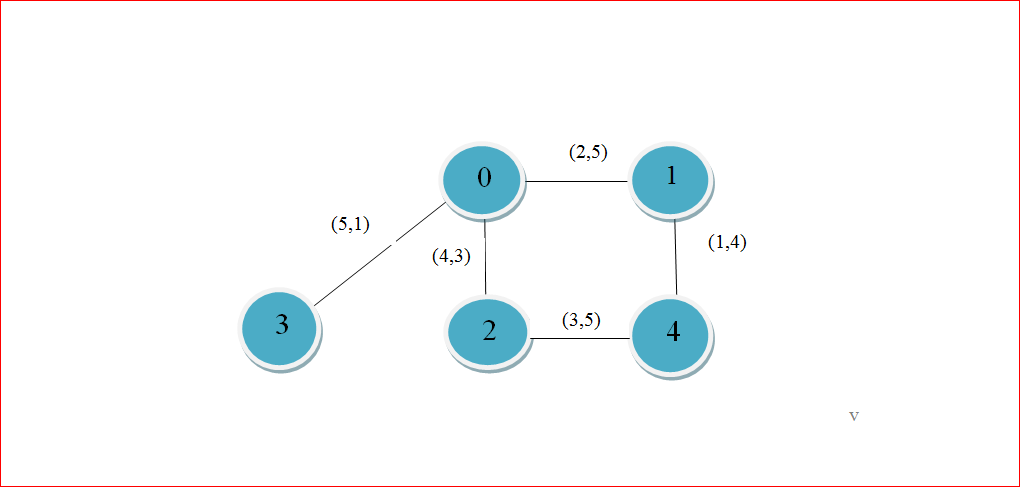
Distance between nodes and toll : sample code used for this is to find out shortest path is by dijkstras algorithm

we can use dijikstras algorithm on weighted tree to find minimum cost and minimum path distance as follows:

we have given following graph as input in our program

created graph is :---

Given graph showing distance and toll amount between cities(nodes).



Matrix showing minimum distance and toll amount between different cities(node of graph) and toll amount between them :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 2 | 3 | 4 |
| 0 | (0,0) | (2,5) | (4,2) | (5,1) | (0,0) |
| 1 | (2,5) | (0,0) | (0,0) | (0,0) | (1,4) |
| 2 | (4,2) | (0,0) | (0,0) | (0,0) | (3,3) |
| 3 | (5,1) | (0,0) | (0,0) | (0,0) | (0,0) |
| 4 | (0,0) | (1,4) | (3,3) | (0,0) | (0,0) |

Matrix to know whether pathway is possible through nodes of graph or not :

Adjajency matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 2 | 3 | 4 |
| 0 | 1 | 1 | 1 | 1 | ∞ |
| 1 | 1 | 1 | ∞ | ∞ | 1 |
| 2 | 1 | ∞ | 1 | ∞ | 1 |
| 3 | 1 | ∞ | ∞ | 1 | ∞ |
| 4 | ∞ | 1 | 1 | ∞ | 1 |

Methodology

Algorithm:

**function** Dijkstra(*Graph*, *source*):

2

3 create vertex set Q

4

5 **for each** vertex *v* in *Graph*: *// Initialization*

6 dist[*v*] ← INFINITY *// Unknown distance from source to v*

7 prev[*v*] ← UNDEFINED *// Previous node in optimal path from source*

8 add *v* to *Q* *// All nodes initially in Q (unvisited nodes)*

9

10 dist[*source*] ← 0 *// Distance from source to source*

11

12 **while** *Q* is not empty:

13 *u* ← vertex in *Q* with min dist[u] *// Source node will be selected first*

14 remove *u* from *Q*

15

16 **for each** neighbor *v* of *u*: *// where v is still in Q.*

17 *alt* ← dist[*u*] + length(*u*, *v*)

18 **if** *alt* < dist[*v*]: *// A shorter path to v has been found*

19 dist[*v*] ← *alt*

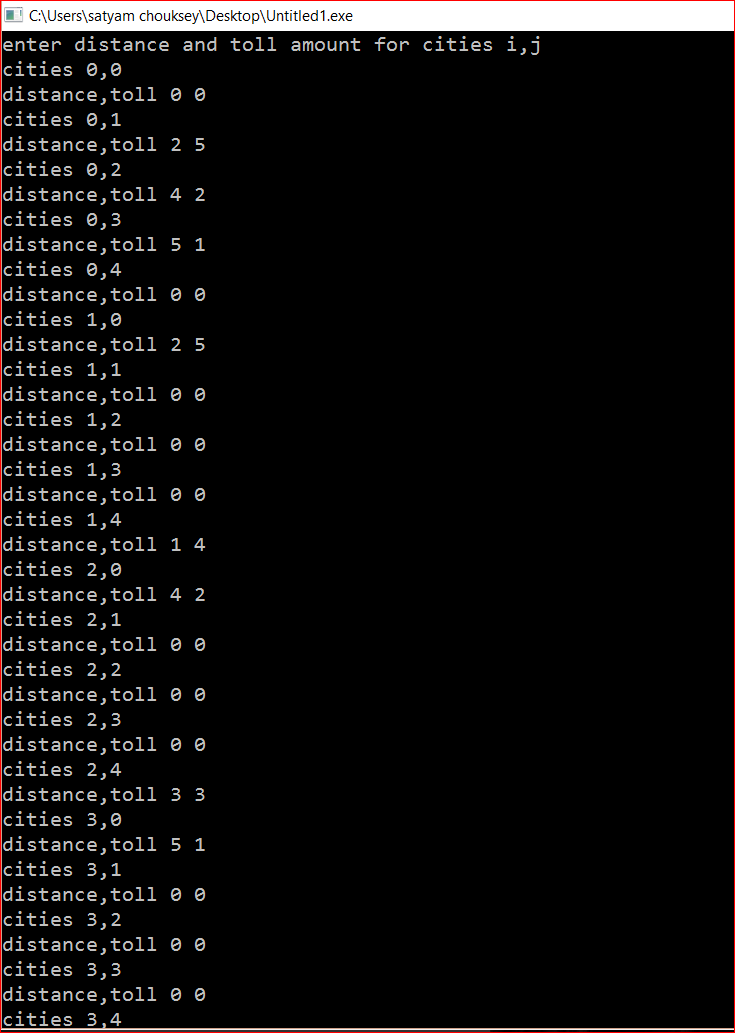
20 prev[*v*] ← *u*

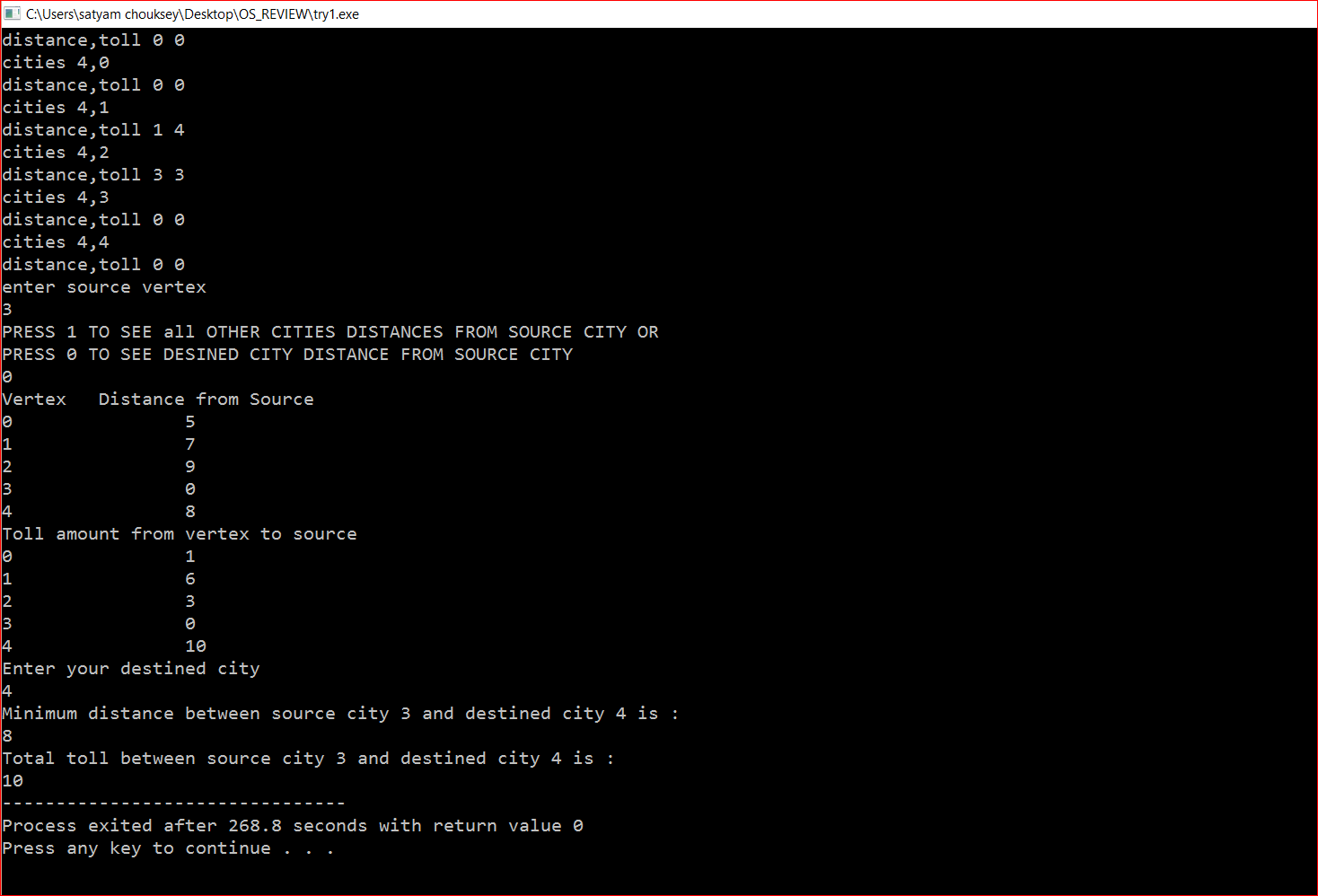
21 **return** dist[], prev[]

* Why do we use Dijkstra’s Algorithm ?
* Algorithms for calculating shortest path from source to sink about as computationally expensive as calculating shortest paths from source to any vertex.

Code for this topic is not given as i am working on its some other applications.

Result:





Conclusion

We successfully simulated code on dijkstras algorithm. We take input of graphs containing distance and toll amount between nodes. Now it will ask for source node. Then program will ask whether to see shortest distance between each node from source node or just want to find shortest distance between source node and destined node. Finally it will ask for destined node and then distance between nodes will be printed with corresponding toll amount.

References

Weblinks:

1. https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm

2.http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf

Book:

   1. A Discipline of Programming Author Edsger W. Dijkstra

Acknowledgements

We would like to express special thanks of gratitude to Professor Sendhil Kumar K.S as well as our University who gave us an opportunity to do this wonderful project on the Turing Machine Simulation for unary addition X+Y.