

## Finding The Shortest Path Using Dijkstra's Algorithm

**Name :** Vitala seeta

**Qualification:** Msc, Mphil (mathematics)

**Designation:** Assistant Professor

**Affiliating Institution:** Rishi MS Institute of Engineering and technology for Women (JNTUH)

### ABSTRACT

Graph theory is one of the most important branches of discrete mathematics which is a study of graphs which are mathematical structures that are used to model pair wise relations between objects. Many discrete problems can be analysed using graph theoretic models. Graph theory plays prominent role in computer science and logistics management. The main application of graph theory is to find the shortest path problems, optimal assignment problems, time tabling problems, travelling salesman problems. Roads plays an important role as every day people from various cities, states and villages has to travel to work, schools, business and to transport the goods. In the roadway networks finding the shortest paths between various locations is the major problem. Because of this problem many shortest path algorithms are developed. In roadway network applications finding shortest path is important in city emergency handling as finding the shortest path is important to save the life and drive guiding systems to reduce the travel time, simultaneously to reduce the cost of implementation. The shortest path problem is to find the path between two vertices on a given graph, such that the sum of the weights on its constituent edges is minimized. In this paper for a Dijkstra's algorithm gives a solution for a single source shortest path problems. In this algorithm the shortest path is calculated for a whole network starting from a source vertex (or) initial vertex by finding the upper bound of the distance between two vertices can be calculated in advance on the given transportation network.

**KEYWORDS:** *Graphs, Transportation networks, Shortest Path, Dijkstra's Algorithm*

## 1. INTRODUCTION

In 1735, the problem of Koinsberg bridge is the origin of graph theory. The town of Koinsberg in Prussia was divided in to four sections included the two regions of the Pregel, Kneiph island and the region between the two branches of the pregel. The swiss mathematician Leonard Euler solved this problem which is the first use of graph theory. Graaphs that have number assigned to each edge are called weighted graphs which are widely used in modelling computer networks. Determining of path of least length between two vertices in a network is one such problem. Important application of weighted graphs are to find the shortest totqal length that visits every vertex of a complete graph exactly once which is commonly known as Travelling salesman problem in which our aim is to find an order in which salesman should visit each of the cities on his route exactly once so that he travels the minimum distance to deliver. Many algorithms are developed to find the shortest path between two vertices in a weighted graph. Because of important applications in graph theory, artificial intelligence, computer network and design of transportation systems, the shortest path problems are investigated over many years.

In 1959 a dutch mathematician Edsger Dijkstra developed an algorithm to find the shortest path in undirected weighted graphs where all the weights are positive.

In section 2 related works of graph theory are discucced, In section 3 basic definations of graph theory are given. In section 3 Dijkstra's algorithm is discussed with examples

## 2. RELATED WORKS

In public transportation systems, the best-path problems find the nature of transfer is that it requires extra costs from an edge to its adjacent edge. In data model transportation problem use of adjacent edges weighted directed graphs. Today, a non-trivial task is to find is to find a good route from one location to other with increased traffic and a road network which is more complex. By using routing algorithms travelling distance and transportation cost can be reduced significantly.

## 3. GRAPH THEORY

A graph  $G = (V, E)$  consists of  $V$ , a nonempty set of *vertices (or nodes)* and  $E$ , a set of *edges*. Each edge has either one or two vertices associated with it which are called end points.

An edge is said to *connect* its end points. A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a *simple graph*. A *directed graph* (or *digraph*) consists of a nonempty set of vertices  $V$  and a set of directed edges  $E$ . The *undirected graphs* consists of set of undirected edges. Graphs that have a weight assigned to each edge are called *weighted graphs*. A *path* is a sequence of edges that begins at a vertex to vertex along the edges of the graph. The *length* of a path in a weighted graph is the sum of the weights of the edges of the path.

#### 4. DIJKSTRA'S ALGORITHM

Dijkstra's algorithm have a series of iterations where labelling procedure is carried at each stage. In the labelling procedure a vertex  $V$  is labelled with the length of the shortest path from initial vertex to  $v$  that contains only vertices which are present in the distinguished set. The vertex which is added to this distinguished set is the vertex with a minimal label among those vertices not already in the set. At each stage of the iteration each vertex  $V$  gets either permanent label denoted by **PL** which is the length  $L_v$  of ma shortest path or a temporary lable **TL** which is upper bound  $L'_v$  for the length of the shortest path.

##### STEPS FOR DIJKSTRA'S ALGORITHM:

Let  $G = (V, E)$  with the vertices  $\{1, 2, 3, \dots, n\}$  and the edges  $(i, j)$  having the lengths  $L_{ij} > 0$ . The steps to find the shortest path from vertex 1 to the vertices  $2, \dots, n$  is given as

**STEP 1:** For the initial vertex give  $L_1 = 0$

Permanent label **PL** = {initial vertex}

For the remaining vertices  $j$  (2, 3, ..., n)

Temporary labelling **TL** =  $L'_j = L_{1j}$

=  $\infty$  If there is no edge from 1 to  $j$  in the graph  $G$

**STEP 2:** Let  $L_k = \min \{ L'_k \} \forall k \in \{2, 3, \dots, n\}$

Find a  $k$  in temporary label **TL** for which  $L_k$  is minimum.

If there are several find the smallest of  $k$

Delete  $k$  from **TL** and include it in **PL**.

**STEP 3:** Join the vertices in permanent label by the corresponding  $k$

**STEP 4:** For all  $j$  in **TL**

Set  $L'_j = \min \{ L'_j, L_k + L_{kj} \}$

Hence the shortest path is obtained.

**Example 1:** To find the shortest path from the vertex  $D$  to all other vertices shown in Figure.1.

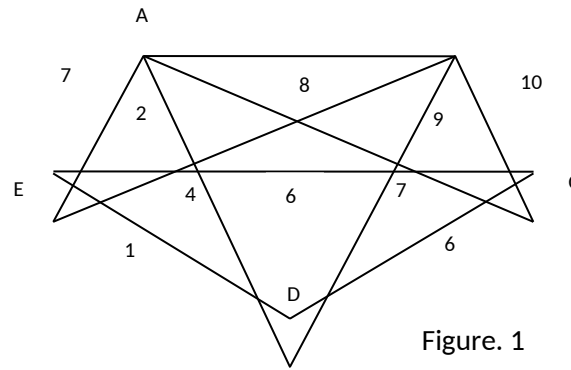


Figure. 1

**Solution:**

The length of weights of the edges are given as  $D-E = 1$ ,  $D-C = 6$ ,  $D-A = 4$ ,  $D-B = 9$ ,  $A-E = 7$ ,  $A-B = 3$ ,  $A-C = 7$ ,  $B-E = 7$ ,  $B-C = 10$ ,  $E-C = 6$ .

Since D is the initial vertex, Hence  $PL = \{ D \}$ .

The length from the initial vertex D to the vertices E, C, A, B are the temporary labelling given by  $L_D = 0$ ,  $L_E = 1$ ,  $L_C = 6$ ,  $L_A = 4$ ,  $L_B = 9$

Temporary label  $TL = \{ E, C, A, B \}$

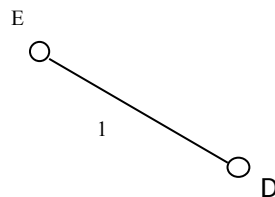
To select the next vertex of shortest length from D.



$$L_2 = \min \{ L_E, L_C, L_A, L_B \} = \min \{ 1, 3, 4, 5 \} = 1 = L_E$$

Therefore the vertex of least weight E is given permanent labelling and remaining vertices are given temporary labelling

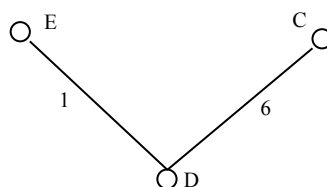
Hence  $PL = \{ D, E \}$ ,  $TL = \{ C, A, B \}$ .



To select the next vertex of shortest length from both D and E .

$$L_3 = \min \{ L_C, L_E + L_{EC} \} = \min \{ 6, 6 + 1 \} = \min \{ 6, 7 \} = 6 = L_C$$

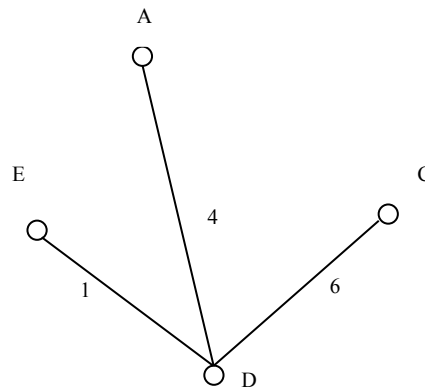
Hence  $PL = \{ D, E, C \}$ ,  $TL = \{ A, B \}$  .



To select the next vertex of shortest length from both D and E and C.

$$L_4 = \min \{ L_A', L_E + L_{EA}, L_C + L_{CA} \} = \min \{ 4, 1+7, 6+3 \} = \min \{ 4, 8, 9 \} = 4 = L_A'$$

Hence  $PL = \{ D, E, C, A \}$ ,  $TL = \{ B \}$



To select the next vertex of shortest length from both D and E and C and A

$$L_5 = \min \{ L_B', L_E + L_{EB}, L_C + L_{CB}, L_A + L_{AC} \} = \min \{ 9, 1+2, 6+10, 4+3 \} = \min \{ 9, 3, 16, 7 \} = 3 = L_E + L_{EB}$$

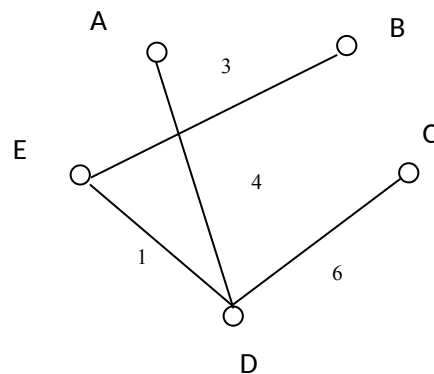


Figure. 1.1

Hence Figure 1.1 gives the shortest path from D to all the vertices A, B, C, E

**Example 2 :** To find the shortest path from Camden to all the cities Trenton, Woodbridge, Asbury Park, Atlantic City, Capemay in Figure 2.

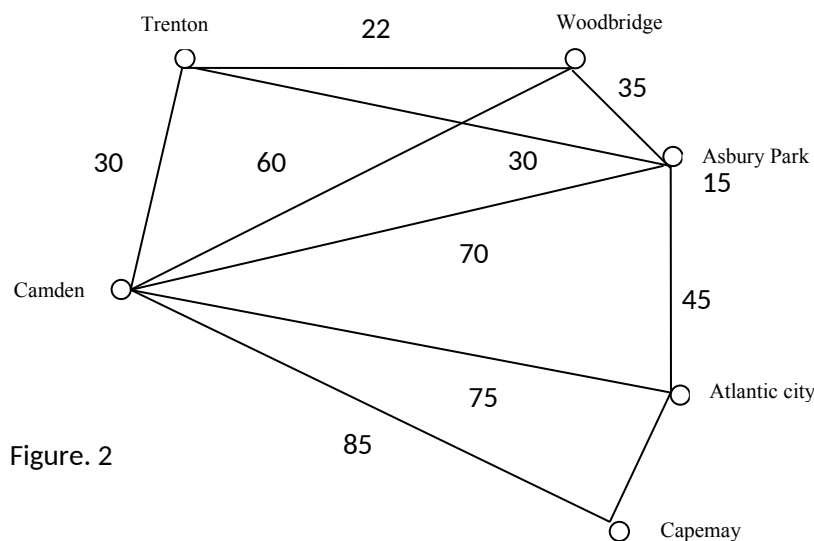


Figure. 2

**Solution:**

The length of weights of the edges are given as Camden--- Trenton = 30, Camden--- Woodbridge = 60, Camden--- Asbury Park = 70, Camden--- Atlantic city = 75, Camden--- Capemay = 85, Trenton--- Woodbridge = 22, Trenton--- Asbury Park = 30, Woodbridge--- Asbury Park = 35, Asbury Park--- Atlantic city = 15, Atlantic city--- Capemay = 45.

Since the initial vertex is Camden

Hence  $PL = \{ \text{Camden} \}$ .

The length from the initial vertex Camden to Trenton, Woodbridge, Asbury Park, Atlantic city, Capemay the vertices are the temporary labelling given by  $L_{\text{Trenton}} = 30$ ,  $L_{\text{Woodbridge}} = 60$ ,  $L_{\text{Asbury Park}} = 70$ ,  $L_{\text{Atlantic city}} = 55$ ,  $L_{\text{Capemay}} = 85$ .

Temporary label  $TL = \{ \text{Trenton}, \text{Woodbridge}, \text{Asbury Park}, \text{Atlantic City}, \text{Capemay} \}$

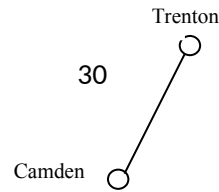
Camden ○

To select the next city of shortest length from Camden

$$L_2 = \min \{ L_{\text{Trenton}}, L_{\text{Woodbridge}}, L_{\text{Asbury Park}}, L_{\text{Atlantic city}}, L_{\text{Capemay}} \} = \min \{ 30, 60, 70, 55, 85 \} = 30 = L_{\text{Trenton}}$$

Therefore the city of least weight Trenton is given permanent labelling and remaining vertices are given temporary labelling

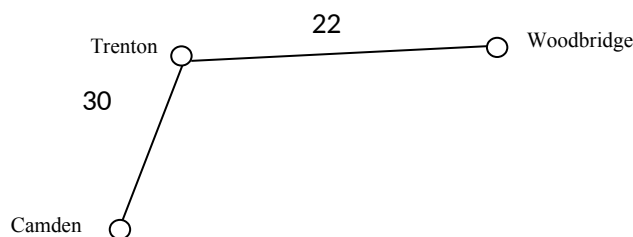
Hence  $PL = \{ \text{Camden}, \text{Trenton} \}$ ,  $TL = \{ \text{Woodbridge}, \text{Asbury Park}, \text{Atlantic City}, \text{Capemay} \}$ .



To select the next city of shortest length from both Camden and Trenton .

$$L_3 = \min \{ L_{\text{Woodbridge}}, L_{\text{Trenton}} + L_{\text{Trenton Woodbridge}} \} = \min \{ 60, 30 + 22 \} = \min \{ 60, 52 \} = 52 = L_{\text{Trenton}} + L_{\text{Trenton Woodbridge}}$$

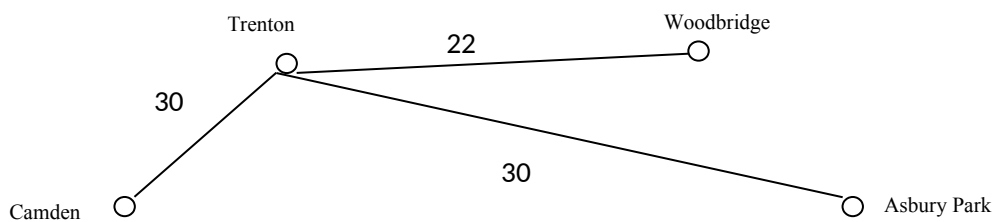
Hence  $PL = \{ \text{Camden}, \text{Trenton}, \text{Woodbridge} \}$ ,  $TL = \{ \text{Asbury Park}, \text{Atlantic City}, \text{Capemay} \}$ .



To select the next city of shortest length from both Camden and Trenton and Woodbridge.

$$L_4 = \min \{ L_{\text{AsburyPark}}, L_{\text{Trenton}} + L_{\text{Trenton Woodbridge}} + L_{\text{Woodbridge AsburyPark}}, L_{\text{CamdenTrenton}} + L_{\text{Trenton AsburyPark}} \} = \min \{ 70, 30 + 22 + 35, 30 + 30 \} = \min \{ 70, 87, 60 \} = 60 = L_{\text{Trenton}} + L_{\text{Trenton Asbury park}}$$

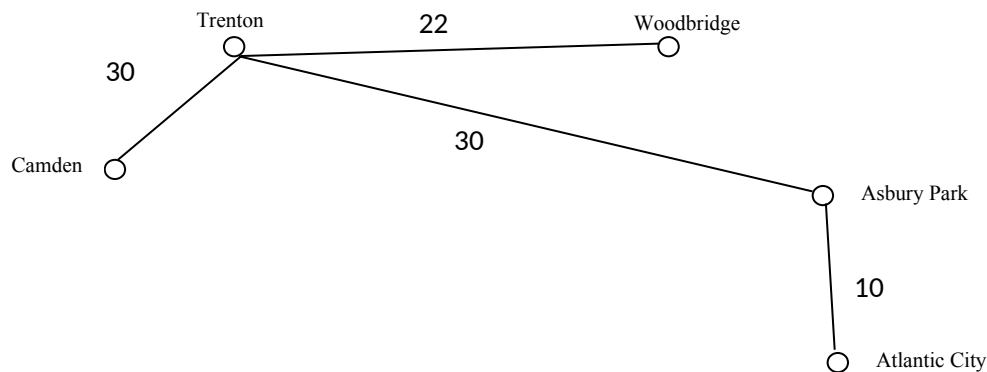
Hence  $PL = \{ \text{Camden}, \text{Trenton}, \text{Asbury Park}, \text{Woodbridge} \}$ ,  $TL = \{ \text{Atlantic City}, \text{Capemay} \}$ .



To select the next city of shortest length from both Camden and Trenton and Woodbridge and Asbury Park.

$$L_5 = \min \{ L_{\text{Atlanticcity}}, L_{\text{Trenton}} + L_{\text{Trenton Woodbridge}} + L_{\text{Woodbridge AsburyPark}} + L_{\text{AsburyParkAtlantic City}}, L_{\text{CamdenTrenton}} + L_{\text{Trenton AsburyPark}} + L_{\text{AsburyPark Atlantic city}} \} = \min \{ 75, 30 + 22 + 35 + 15, 30 + 30 + 10 \} = \min \{ 75, 102, 70 \} = 70 = L_{\text{Trenton}} + L_{\text{Trenton Asbury park}} + L_{\text{AsburyPark Atlantic city}}$$

Hence  $PL = \{ \text{Camden}, \text{Trenton}, \text{Asbury Park}, \text{Atlantic City}, \text{Woodbridge} \}$ ,  $TL = \{ \text{Capemay} \}$ .



To select the next city of shortest length from both Camden and Trenton and Woodbridge and Asbury Park and Atlantic city.

$$L_6 = \min \{ L_{\text{Capemay}'}, L_{\text{Trenton}} + L_{\text{trenton AsburyPark}} + L_{\text{AsburyPark Atlantic city}} + L_{\text{Atlantic City Capemay.}} \} = \min\{85, 30+30+10+45\} = \min \{ 85, 115\} = 85 \{ L_{\text{Capemay}'} \}$$

Hence  $PL = \{ \text{Camden}, \text{Trenton}, \text{Asbury Park}, \text{Atlantic City}, \text{Woodbridge}, \text{Capemay} \}$ ,  $TL = \{0\}$ .

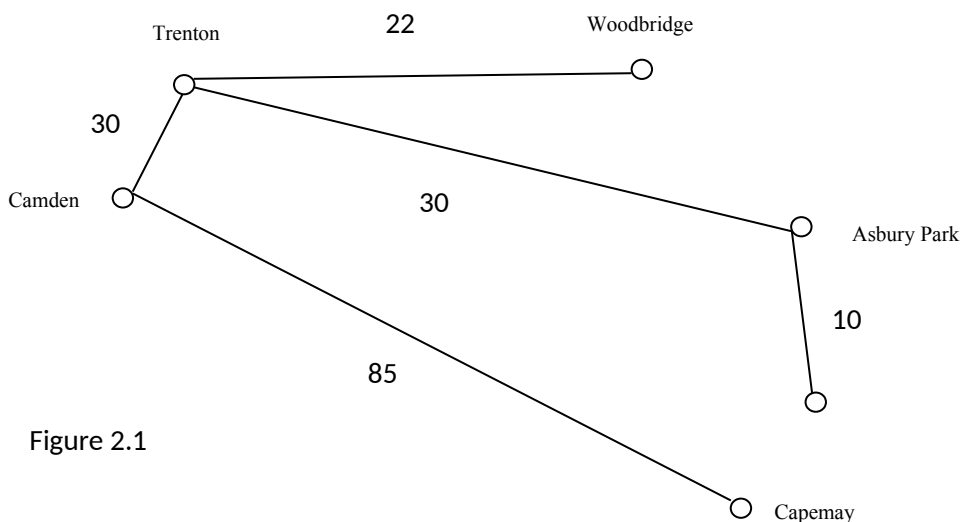


Figure 2.1

Hence Figure 2.1 displays the shortest path from Camden to all the cities.

**Conclusion:** Finding the shortest path for the graph problems is difficult and interesting. Hence a practical algorithm for the shortest path problem in transportation problems is proposed. The limitation of this model is the weights of the graph are positive and graph



should be connected to find the shortest path. Among the problems presented have realistic approach in our life experiences. This algorithm can be applied everywhere in road building, travelling and paddlers. There are several algorithms to find the shortest path in the graph theory.

### References:

1. Aghaei, M. R. S., Zukarnain, Z. A., Mamat, A., & Zainuddin, H. (2008, December). A *Hybrid Algorithm for the Shortest-path Problem in the Graph*. In 2008 International Conference on Advanced Computer Theory and Engineering (pp. 251-255). IEEE.
2. Dobrilovic, D., Jevtic, V., Beker, I., & Stojanov, Z. (2012, May). *Shortest-path based model for warehouse inner transportation optimization*. In 2012 7th International Symposium on Applied Computational Intelligence and Informatics (SACI)(pp. 63-68). IEEE.
3. Jigang, W., Han, P., Jagadeesh, G. R., & Srikanthan, T. (2010). Practical algorithm for shortest path on large networks with time-dependent edge-length. In 2010 2nd International Conference on Computer Engineering and Technology.
4. Wu, H., Marshall, A., & Yu, W. (2007, July). *Path planning and following algorithms in an indoor navigation model for visually impaired*. In Second International Conference on Internet Monitoring and Protection (ICIMP 2007) (pp. 38-38). IEEE.
5. Wang, H., Hu, M., & Xiao, W. (2010, March). *A new public transportation data model and shortest-path algorithms*. In 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010) (Vol. 1, pp. 456-459). IEEE.
6. Xiao, J. X., & Lu, F. L. (2010, February). *An improvement of the shortest path algorithm based on Dijkstra's algorithm*. In 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE) (Vol. 2, pp. 383-385). IEEE.
7. Zhan, F. B., & Noon, C. E. (1998). *Shortest path algorithms: an evaluation using real road networks*. *Transportation science*, 32(1), 65-73.
8. Almash Alam, Md., & Omar Faruq, Md. *Finding the shortest path for road networks using Dijkstra's Algorithm* Vol. 1, No. 2 ( 2019) 2687-8518.
9. Badwaik Jyothi,S. *Recent advances in graph theory and its applications*. International Research journal of Science and Engineering (2020) 2322-0015.
10. Sanjay Kumar Bisen. *Application of Graph Theory in Operations Research*. Vol. 2 No. 5(2017) 2456-2165