

LABORATORY REPORT

**Algorithm Laboratory (CS-39001)**

**B.Tech Program in ECS**

Submitted By

**Name:- SANNIDHI DEB**

**Roll No: 2330044**

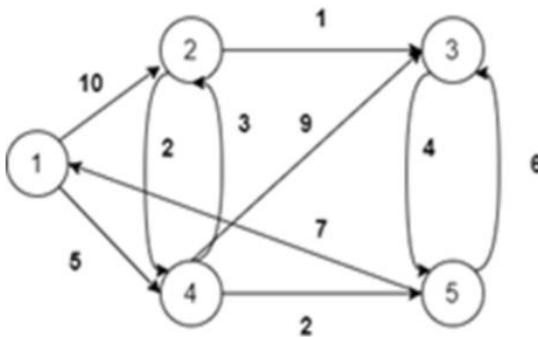


**Kalinga Institute of Industrial Technology  
(Deemed to be University) Bhubaneswar, India**

Autumn, 2025

## **Table of Contents**

#	Title	Date of Experiment	Date of Submission	Remarks
1.	Revision of Data Structures	24/07/2025	06/08/2025	
2.	Design and Analysis of Fundamental Algorithms	07/08/2025	20/08/2025	
3.	Divide and Conquer Method	14/08/2025	20/08/2025	
4.	Heap Sort	21/08/2025	27/08/2025	
5.	Greedy Algorithm	18/09/2025	24/09/2025	
6.	Huffman Coding	01/10/2025	10/10/2025	
7.	Minimum Spanning Tree	01/10/2025	10/10/2025	
8.	Single Source Shortest Path	09/10/2025	15/10/2025	

<b>Experiment Number</b>	8.1																								
	<p>Given a directed graph G (V, E) and a starting vertex 's'.</p> <ul style="list-style-type: none"> <li>• Determine the lengths of the shortest paths from the starting vertex 's' to all other vertices in the graph G using Dijkstra's Algorithm.</li> <li>• Display the shortest path from the given source 's' to all other vertices.</li> </ul>  <p><b>Input:</b>  Enter the Number of Vertices: 5  Enter the Source Vertex: 1</p> <p><b>Output:</b></p> <table border="1"> <thead> <tr> <th>Source</th> <th>Destination</th> <th>Cost</th> <th>Path</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>0</td> <td>-</td> </tr> <tr> <td>1</td> <td>2</td> <td>8</td> <td>1-&gt;4-&gt;2</td> </tr> <tr> <td>1</td> <td>3</td> <td>9</td> <td>1-&gt;4-&gt;2-&gt;3</td> </tr> <tr> <td>1</td> <td>4</td> <td>5</td> <td>1-&gt;4</td> </tr> <tr> <td>1</td> <td>5</td> <td>7</td> <td>1-&gt;4-&gt;5</td> </tr> </tbody> </table>	Source	Destination	Cost	Path	1	1	0	-	1	2	8	1->4->2	1	3	9	1->4->2->3	1	4	5	1->4	1	5	7	1->4->5
Source	Destination	Cost	Path																						
1	1	0	-																						
1	2	8	1->4->2																						
1	3	9	1->4->2->3																						
1	4	5	1->4																						
1	5	7	1->4->5																						
<b>Date of Experiment</b>	09/10/2025																								
<b>Date of Submission</b>	15/10/2025																								

## 1. Algorithm:-

START

sannidhi Deb, 2330049

1. Input: Number of vertices 'n', source vertex 's'  
cost adjacent matrix

2. Initialize:

i)  $\text{dist}[i] = \text{cost}[s][i]$  for all vertices 'i'

ii)  $\text{visited}[i] = 0$

iii)  $\text{dist}[s] = 0$

3. Repeat for all vertices:

i) select unvisited vertex 'u' with smallest  $\text{dist}[u]$

ii) for each neighbour 'v' of 'u', if  $\text{dist}[v] > \text{dist}[u] + \text{cost}[u][v]$   
then update  $\text{dist}[v] = \text{dist}[u] + \text{cost}[u][v]$  and record predecessor.

4. Display the shortest path and distance from 's' to every vertex

END

## 2. Code:-

```
#include <stdio.h>
#define INF 999
void dijkstra(int n, int cost[10][10], int source) {
    int dist[10], visited[10], parent[10];
    int count, mindist, nextnode, i, j;
    for (i = 1; i <= n; i++) {
        dist[i] = cost[source][i];
        visited[i] = 0;
```

```

parent[i] = source;
}

dist[source] = 0;
visited[source] = 1;
count = 1;

while (count < n - 1) {
    mindist = INF;
    for (i = 1; i <= n; i++) {
        if (dist[i] < mindist && !visited[i]) {
            mindist = dist[i];
            nextnode = i;
        }
    }
    visited[nextnode] = 1;
    for (i = 1; i <= n; i++) {
        if (!visited[i])
            if (mindist + cost[nextnode][i] < dist[i]) {
                dist[i] = mindist + cost[nextnode][i];
                parent[i] = nextnode;
            }
    }
    count++;
}

printf("\nSannidhi Deb\n 2330044\n\n");
printf("\nSource\tDestination\tCost\tPath");
for (i = 1; i <= n; i++) {
    if (i != source) {

```

```
printf("\n%d\t%d\t%d\t%d", source, i, dist[i], i);

j = i;

while (j != source) {

    j = parent[j];

    printf("<-%d", j);

}

}

printf("\n");

}

int main() {

int n = 5, source = 1;

int cost[10][10] = {

{0,0,0,0,0,0},

{0,0,10,INF,5,INF},

{0,INF,0,1,2,INF},

{0,INF,INF,0,INF,4},

{0,INF,3,9,0,2},

{0,INF,INF,6,INF,0}

};

dijkstra(n, cost, source);

return 0;

}
```

### 3. Results/Output:- Entire Screen Shot including Date & Time:-

The screenshot shows a terminal window with the following content:

```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>gcc exp8_1.c -o exp8_1
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>exp8_1

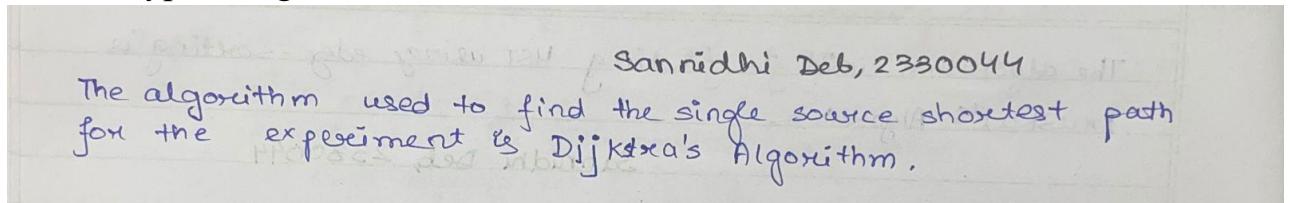
Sannidhi Deb
2330044

Source Destination Cost Path
1 2 8 2<-4<-1
1 3 9 3<-2<-4<-1
1 4 5 4<-1
1 5 7 5<-4<-1
```

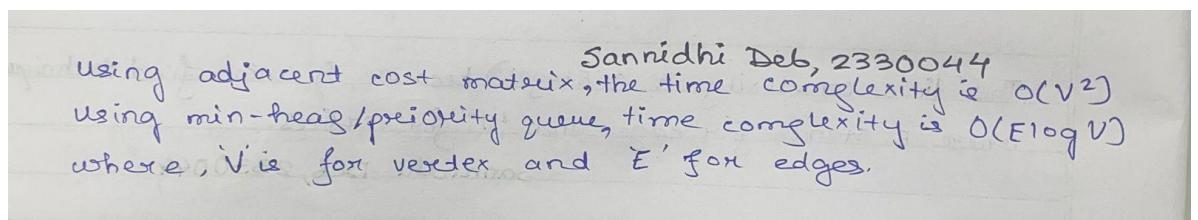
The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), PORTS, and SPELL CHECKER. There are 7 notifications in the spell checker tab. The status bar at the bottom shows the current directory as C:\Users\debsa\OneDrive\Desktop\AL\_Lab\_044, line 38, column 45, spaces: 4, encoding: UTF-8, line feed: LF, file type: C, and other system information like battery level, language (ENG IN), and date/time (12:24, 09-10-2025).

### 4. Remarks:-

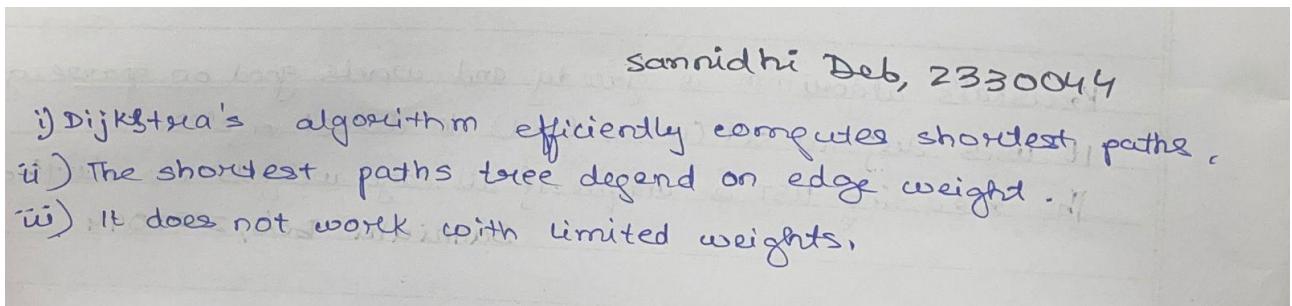
1. What type of algorithm is used?



2. Analyze the complexity of your algorithm.



### 3. Any other observations?



**5. Conclusion:-** Dijkstra's Algorithm successfully determines the shortest path and cost from a given source vertex to all other vertices in a directed weighted graph.  
It is one of the most efficient algorithms for single-source shortest path problems with non-negative edge weights.

Sannidhi Deb  
(2330044)

Signature of the FIC

Sannidhi Deb

(Name of the FIC)

