

LABORATORY REPORT

Algorithm Laboratory (CS-39001)

B.Tech Program in ECSc

Submitted By

Name:- SANNIDHI DEB

Roll No: 2330044



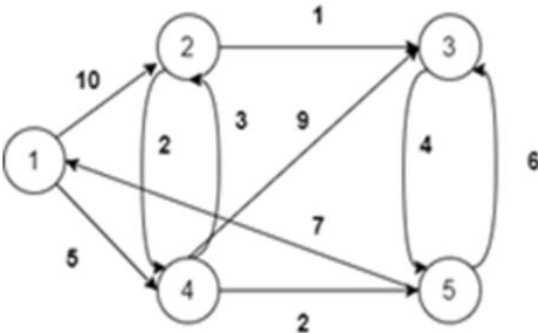
Kalinga Institute of Industrial Technology

(Deemed to be University) Bhubaneswar, India

Autumn, 2025

Table of Contents

[illegible]

Experiment Number	8.1																								
Experiment Title	<p>Given a directed graph $G(V, E)$ and a starting vertex 's'.</p> <ul style="list-style-type: none">● Determine the lengths of the shortest paths from the starting vertex 's' to all other vertices in the graph G using Dijkstra's Algorithm.● Display the shortest path from the given source 's' to all other vertices. <div></div> <p>Input:</p> <p>Enter the Number of Vertices: 5</p> <p>Enter the Source Vertex: 1</p> <p>Output:</p> <table><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->4->2</td></tr><tr><td>1</td><td>3</td><td>9</td><td>1->4->2->3</td></tr><tr><td>1</td><td>4</td><td>5</td><td>1->4</td></tr><tr><td>1</td><td>5</td><td>7</td><td>1->4->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																						
Date of Experiment	09/10/2025																								
Date of Submission	15/10/2025																								

1. Algorithm:-

START Sannidhi Deb, 2330049

1. Input: Number of vertices 'n', source vertex 's'
cost adjacent matrix
2. Initialise:
 - 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\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}

    };

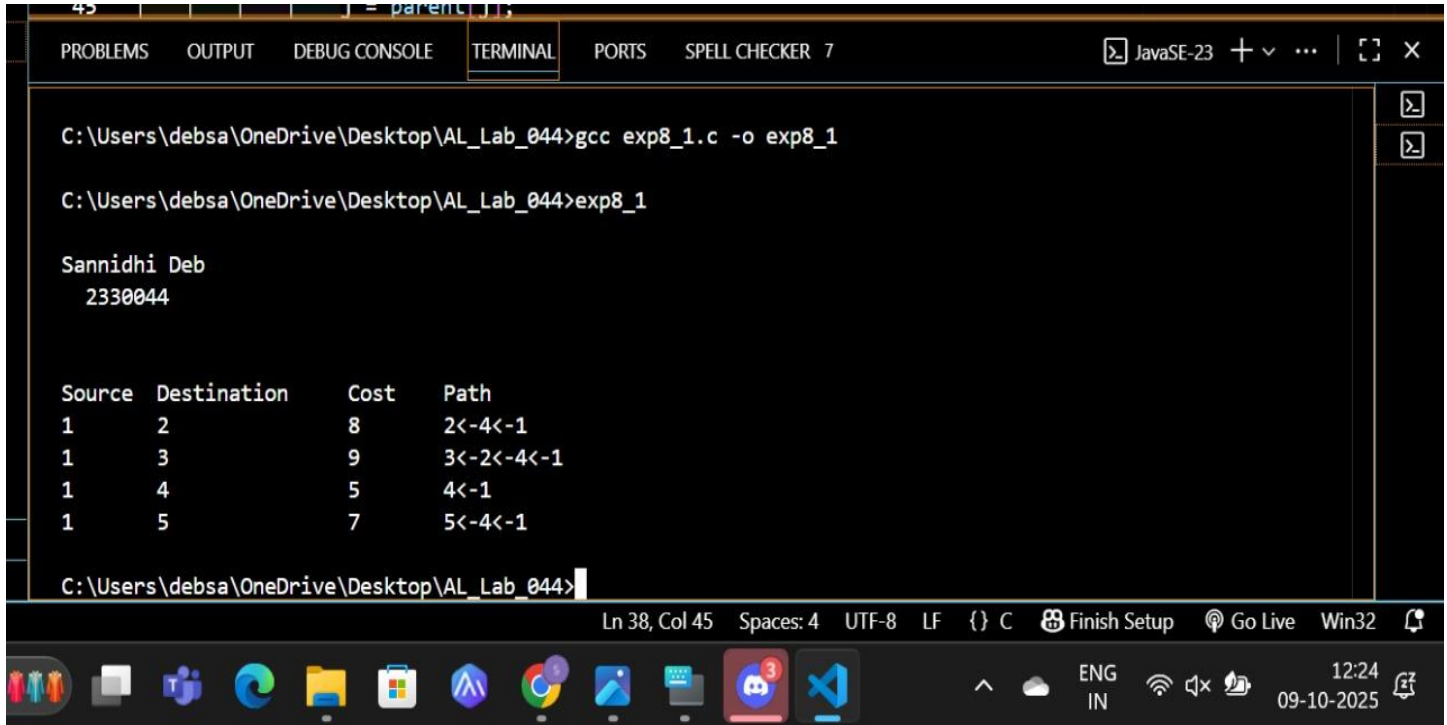
    dijkstra(n, cost, source);

    return 0;

}

```

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



```
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
```

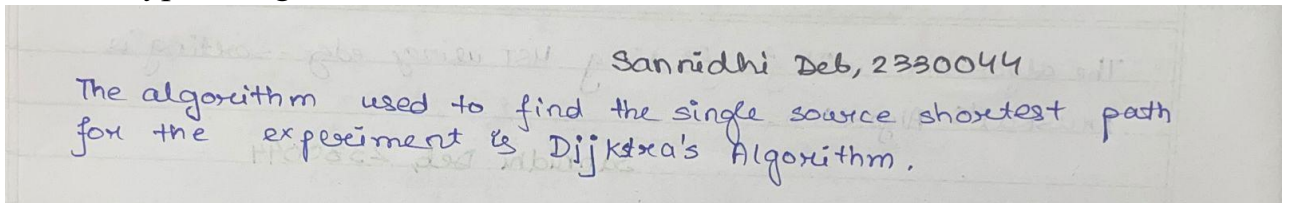
Source	Destination	Cost	Path
1	2	8	2<-4<-1
1	3	9	3<-2<-4<-1
1	4	5	4<-1
1	5	7	5<-4<-1

```
C:\Users\debsa\OneDrive\Desktop\AL_Lab_044>
```

Ln 38, Col 45 Spaces: 4 UTF-8 LF {} C Finish Setup Go Live Win32 12:24 09-10-2025

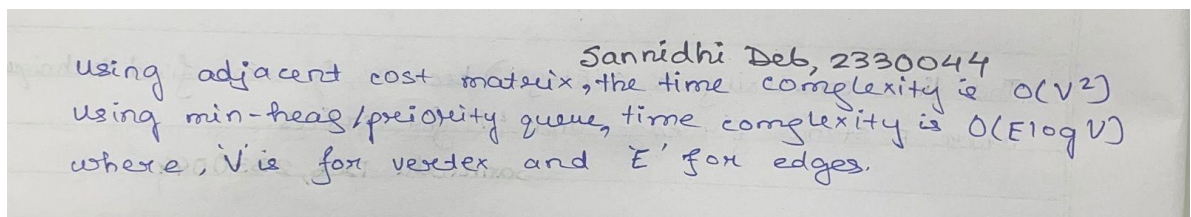
4. Remarks:-

1. What type of algorithm is used?



Sannidhi Deb, 2330044
The algorithm used to find the single source shortest path for the experiment is Dijkstra's Algorithm.

2. Analyze the complexity of your algorithm.



Sannidhi Deb, 2330044
using adjacent cost matrix, the time complexity is $O(V^2)$
using min-heap/priority queue, time complexity is $O(E \log V)$
where, V is for vertex and E for edges.

3. Any other observations?

Sannidhi Deb, 2330044

- i) Dijkstra's algorithm efficiently computes shortest paths.
- ii) The shortest paths tree depend on edge weight.
- iii) It does not work with limited weights.

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)

Sannidhi Deb

Signature of the FIC

(Name of the FIC)

