

CYCLE-2

LAB-3: Implement Dijkstra's algorithm to compute the shortest path for a given topology.

Program:

```
#include <iostream>
using namespace std;
int a[30][30], source, dist[30], path[30];

void dijkstar(int a[][30], int n)
{
    int visited[n];
    for (int i = 0; i < n; i++)
    {
        dist[i] = a[source][i];
        path[i] = source;
        visited[i] = 0;
    }
    visited[source] = 1;
    for (int c = 0; c < n; c++)
    {
        int min = 999, u;
        for (int j = 0; j < n; j++)
        {
            if (dist[j] < min && visited[j] != 1)
            {
                min = dist[j];
                u = j;
            }
        }
        visited[u] = 1;
        for (int i = 0; i < n; i++)
        {
            if (min + a[u][i] < dist[i])
            {
                dist[i] = min + a[u][i];
                path[i] = u;
            }
        }
    }
}

int main()
{
    int n;
    cout << "Enter the no. of vertices : " << endl;
    cin >> n;
    cout << "Enter the adjacency matrix(Enter 9999 for infinity): " << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> a[i][j];
        }
    }
}
```

```

cout << "Enter the source vertex : " << endl;
cin >> source;
cout << "The shortest paths from vertex ' " << source << " ' are : " << endl;
cout << "Vertex paths" << endl;
dijkstra(a, n);
for (int i = 0; i < n; i++)
{
    int k = i;
    while (k != source)
    {

        cout << k << " <- ";
        k = path[k];
    }
    cout << source << " = ";
    cout << "Path cost:" << dist[i] << endl;
}
return 0;
}

```

Output:

```

Enter the no. of vertices :
5
Enter the adjacency matrix(Enter 9999 for infinity):
0 10 9999 9999 6
9999 0 1 9999 2
9999 9999 0 5 9999
6 9999 7 0 9999
9999 3 9 2 0
Enter the source vertex :
1
The shortest paths from vertex ' 1 ' are :
Vertex paths
0 <- 3 <- 4 <- 1 = Path cost:10
1 = Path cost:0
2 <- 1 = Path cost:1
3 <- 4 <- 1 = Path cost:4
4 <- 1 = Path cost:2

Process returned 0 (0x0)   execution time : 35.891 s
Press any key to continue.

```