

Lab Program 9

Implement Dijkstra Algorithm to find shortest path

```

#include <bits/stdc++.h>
using namespace std;
int a[30][30], n;

int minimum(int visited[], int dist[])
{
    int mindis = 10000, mini;
    for (int i = 0; i < n; i++)
    {
        if (!visited[i] && dist[i] < mindis)
        {
            mindis = dist[i];
            mini = i;
        }
    }
    return mini;
}

void dijkstra(int src)
{
    int dist[n], visited[n];
    for (int i = 0; i < n; i++)
    {
        dist[i] = 10000;
        visited[i] = 0;
    }
    dist[src] = 0;
    for (int i = 0; i < n - 1; i++)
    {
        int u = minimum(visited, dist);
        visited[u] = 1;
        for (int v = 0; v < n; v++)
        {
            if (!visited[v] && a[u][v] != 10000 &&
                dist[u] + a[u][v] < dist[v])
            {
                dist[v] = dist[u] + a[u][v];
            }
        }
    }
}

```

p

```

    dist[v] = dist[u] + a[u][v];
}

```

```

}
cout << "Shortest path to all other vertices from " << src << endl;
cout << "is " << endl;
cout << "Vertex \t Distance from source" << endl;
for (int i = 0; i < n; i++)
{
    if (i != src)
        cout << i << "\t\t" << dist[i] << endl;
}
}

```

```

int main()
{
    cout << "Enter no of vertices " << endl;
    cin >> n;
    cout << "Enter the weighted adjacency matrix " << endl;
    (Enter 10000 if there is no edge) << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
            cin >> a[i][j];
    }

    int src;
    cout << "Enter source vertex " << endl;
    cin >> src;
    dijkstra(src);
    return 0;
}

```