Q. Represent a graph of city using adjacency matrix /adjacency list. Nodes should represent the various landmarks and links should represent the distance between them. Find the shortest path using Dijkstra's algorithm from single source to all destination. Analyse the implemented algorithm for space and time complexity

Code

```
#include <iostream>
#include <limits.h>

using namespace std;

#define V 9

class Dijkstra {
    private:
    int distance[V];
    bool shortest_path[V];
    int parent[V];

    public:
    int minDistance(int distance[], bool shortest_path[]);
    void printShortestPath(int distance[]);
    void findSPT(int graph[V][V], int source);
};

int Dijkstra ::minDistance(int distance[], bool shortest_path[]){
    int min = INT_MAX;
    int min_index;

    for(int v=0;v<V; v++){
        if (shortest_path[v]== false && distance[v]<=min){
            min = distance[v];
            min_index = v;
        }
    }
    return min_index; //returns closest vertex
}

void Dijkstra ::findSPT(int graph[V][V], int source){
    for (int i =0; i<V; i++){
        distance[i] = INT_MAX;
        shortest_path[i] = false;
    }

    distance[source]=0; //distance of source vertex from itself is zero

    for (int count =0; count <V-1; count++){
        int u = minDistance(distance, shortest_path);
        shortest_path[u] = true;

        for (int v=0; v<V; v++){
```

```
            if ((graph[u][v]>0) && (graph[u][v]+
distance[u]<distance[v]) &&shortest_path[v]== false && (distance[u] !=
INT_MAX)){
                distance[v] = distance[u] + graph[u][v];
            }
        }
    }
    printShortestPath(distance);
}
void Dijkstra :: printShortestPath(int distance[])
{
    cout <<"Vertex \t Distance from Source" << endl;
    for (int i = 0; i < V; i++)
        cout  << i << " \t\t"<<distance[i]<< endl;
}

int main(){
    Dijkstra spt;
     int graph[V][V] = { { 0, 4, 0, 0, 0, 0, 0, 8, 0 },
                         { 4, 0, 8, 0, 0, 0, 0, 11, 0 },
                         { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
                         { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
                         { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
                         { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
                         { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
                         { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
                         { 0, 0, 2, 0, 0, 0, 6, 7, 0 } };

    spt.findSPT(graph, 0);
}
```

Sample Output

```
Dijkstra's Algorithm to find Shortest Path

Vertex   Distance from Source
0                0
1                4
2                12
3                19
4                21
5                11
6                9
7                8
8                14
PS C:\Users\suhas\OneDrive\Desktop\DSAL\.vscode\graphs>
```