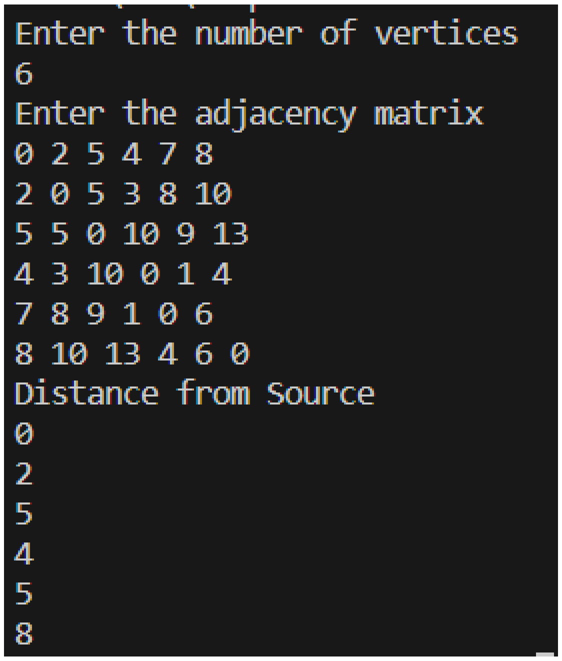
Dijkstra’s Algorithm

Perumalla Dharan
AP21110010201

```
// AP21110010201
// Perumalla Dharan

#include <iostream>
#include <climits>
using namespace std;
int minDistance(int *dist,bool *visited,int n){
    int min = INT_MAX,min_index;
    for(int i=0;i<n;i++){
        if(!visited[i] && dist[i]<=min){
            min = dist[i];
            min_index = i;
        }
    }
    return min_index;
}
void dijkstra(int **graph,int n){
    int *dist = new int[n];
    bool *visited = new bool[n];
    for(int i=0;i<n;i++){
        dist[i] = INT_MAX;
        visited[i] = false;
    }
    dist[0] = 0;
    for(int i=0;i<n-1;i++){
        int u = minDistance(dist,visited,n);
        visited[u] = true;
        for(int j=0;j<n;j++){
            if(!visited[j] && graph[u][j] && dist[u]!=INT_MAX &&
dist[u]+graph[u][j]<dist[j]){
                dist[j] = dist[u]+graph[u][j];
            }
        }
    }
    cout<<"Distance from Source"<<endl;
    for(int i=0;i<n;i++){
        cout<<dist[i]<<endl;
    }
}
int main(){
    int n;
    cout<<"Enter the number of vertices"<<endl;
    cin>>n;
    int **graph = new int*[n];
    for(int i=0;i<n;i++){
        graph[i] = new int[n];
    }
    cout<<"Enter the adjacency matrix"<<endl;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            cin>>graph[i][j];
        }
    }
    dijkstra(graph,n);
    return 0;
}
```

Output:



```
Enter the number of vertices
6
Enter the adjacency matrix
0 2 5 4 7 8
2 0 5 3 8 10
5 5 0 10 9 13
4 3 10 0 1 4
7 8 9 1 0 6
8 10 13 4 6 0
Distance from Source
0
2
5
4
5
8
```