DCCN LAB 1

Name: AMAN KUMAR GUPTA

Roll no: BTech/25013/18

Branch: CSE

Program

Aim : Write a program to implement Dijkstra's shortest path algorithm.

Code:

```
from collections import defaultdict
import sys

class Graph:
    def minDistance(self,dist,queue):
        minimum = float("Inf")
        min_index = -1
        for i in range(len(dist)):
            if dist[i] < minimum and i in queue:
                minimum = dist[i]
                min_index = i
        return min_index

    def printPath(self, parent, j):
        if parent[j] == -1 :
            print(j,end=" ")
            return
        self.printPath(parent , parent[j])
        print(j,end=" ")

    def printSolution(self, dist, parent,src,dest):
        print(f"Minimum Distance From source {src} to Destination {dest} =
 {dist[dest]}\n")
        print("Path = ",end = " ")
        self.printPath(parent,dest)
        print("\n")

    def dijkstra(self, graph, src, dest):

        row = len(graph)
        col = len(graph[0])
        dist = [float("Inf")] * row
        parent = [-1] * row
        dist[src] = 0
        queue = []
        for i in range(row):
            queue.append(i)
        while queue:
            u = self.minDistance(dist,queue)
            queue.remove(u)
            for i in range(col):
                if graph[u][i] and i in queue:
                    if dist[u] + graph[u][i] < dist[i]:
```

```
                        dist[i] = dist[u] + graph[u][i]
                        parent[i] = u
        self.printSolution(dist,parent,src,dest)


def get_input(path):
    f = open(path,'r')
    txt = f.read()
    txt = txt.split("\n")
    n = len(txt)
    arr = []
    for i in range(n-2):
        temp = txt[i].split(",")
        temp = list(map(int,temp))
        arr.append(temp)
    s = int(txt[n-2])
    d = int(txt[n-1])

    return arr,s,d


graph,source,destination = get_input(sys.argv[1])
print("\nGraph Matrix : ")
print(*graph,sep="\n")
print(f"\nSource = {source}\nDestination = {destination}\n")

g= Graph()
g.dijkstra(graph,source,destination)
```

Input and Output :

a.txt

```
0, 3, 0, 0, 0, 0, 0, 8, 0
3, 0, 6, 0, 0, 0, 0, 17, 0
0, 6, 1, 14, 0, 4, 0, 0, 2
0, 1, 14, 0, 13, 14, 0, 0, 0
0, 0, 0, 13, 0, 15, 0, 0, 0
0, 0, 4, 14, 15, 0, 2, 0, 0
0, 0, 0, 0, 0, 2, 0, 1, 6
8, 17, 0, 0, 0, 0, 1, 0, 7
0, 0, 2, 0, 0, 0, 6, 7, 0
1
5
```

```
PS C:\Users\LENOVO\Desktop\BTech_6_Lab\dccn> & C:/Users/LENOVO/anaconda3/python.exe c:/Users/LENOVO/Desktop/BTech_6_Lab/dccn/test.py a.txt

Graph Matrix :
[0, 3, 0, 0, 0, 0, 0, 8, 0]
[3, 0, 6, 0, 0, 0, 0, 17, 0]
[0, 6, 1, 14, 0, 4, 0, 0, 2]
[0, 1, 14, 0, 13, 14, 0, 0, 0]
[0, 0, 0, 13, 0, 15, 0, 0, 0]
[0, 0, 4, 14, 15, 0, 2, 0, 0]
[0, 0, 0, 0, 0, 2, 0, 1, 6]
[8, 17, 0, 0, 0, 0, 1, 0, 7]
[0, 0, 2, 0, 0, 0, 6, 7, 0]

Source = 1
Destination = 5

Minimum Distance From source 1 to Destination 5 = 10

Path =  1 2 5
```