

Worksheet 4

Student Name: Harsh

UID: 24MCA20045

Branch: MCA(AI&ML)

Section/Group: 1-A

Semester: 1st semester

Date of Performance: 13/03/2025

Subject Name: Design and Analysis of Algorithm Subject Code: 24CAP-612

Aim:

From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

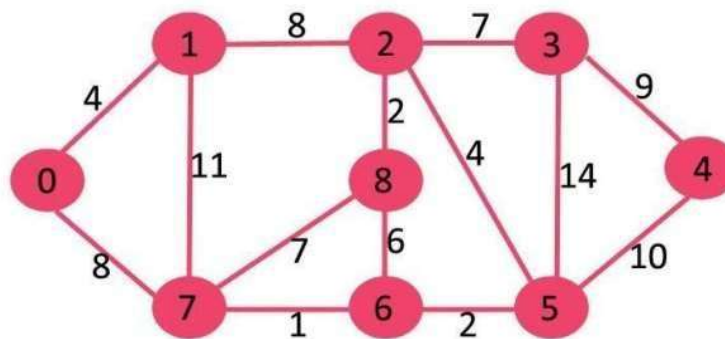


Figure 1: Graph-1

Task To be done:

- Objective: Implement Dijkstra's algorithm to find the shortest paths from a given source vertex (node) to all other vertices in the provided weighted connected graph.
- Steps:
 - o Input the weighted graph (as seen in Figure 1).
 - o Select a source vertex, e.g., vertex 0.
 - o Use Dijkstra's algorithm to compute the shortest paths from the source vertex to every other vertex in the graph.
 - o Display the shortest distances and paths for all vertices.

Source Code:

```
import sys

def min_distance(dist, spt_set, V):
    min_val = sys.maxsize
    min_index = -1

    for v in range(V):
        if not spt_set[v] and dist[v] <= min_val:
            min_val = dist[v]
            min_index = v

    return min_index

def dijkstra(graph, src, V):
    dist = [sys.maxsize] * V
    spt_set = [False] * V

    dist[src] = 0

    for _ in range(V - 1):
        u = min_distance(dist, spt_set, V)
        spt_set[u] = True

        for v in range(V):
            if (not spt_set[v] and graph[u][v] and dist[u] != sys.maxsize
                and dist[u] + graph[u][v] < dist[v]):
                dist[v] = dist[u] + graph[u][v]

    print_solution(dist, V)

def print_solution(dist, V):
    print("Vertex \t Distance from Source")
    for i in range(V):
        print(f'{i} \t {dist[i]}')

if __name__ == "__main__":
    graph = [
        [0, 4, 0, 0, 0, 0, 8, 0],
        [4, 0, 8, 0, 0, 0, 11, 0],
        [0, 8, 0, 7, 0, 4, 0, 2],
        [0, 0, 7, 0, 9, 14, 0, 0],
        [0, 0, 0, 9, 0, 10, 0, 0],
```

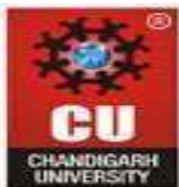
```
[0, 0, 4, 14, 10, 0, 2, 0],  
[8, 11, 0, 0, 0, 2, 0, 1],  
[0, 0, 2, 0, 0, 0, 1, 0]  
]  
src = 0  
dijkstra(graph, src, len(graph))
```

Output:

```
Vertex    Distance from Source  
0         0  
1         4  
2        11  
3        18  
4        20  
5        10  
6         8  
7         9  
  
Process finished with exit code 0
```

Learning Outcome:

- **Understanding Dijkstra's Algorithm:** After implementing and executing this experiment, you will gain a solid understanding of how Dijkstra's algorithm works to find the shortest path in a weighted graph.
- **Graph Representation:** You will learn how to represent graphs using an adjacency matrix and manipulate graph data to implement graph algorithms.
- **Application in Real-World Problems:** You will understand how Dijkstra's algorithm can be applied to various real-world problems, such as finding the shortest path in a network of roads, telecommunications, or computer networks.
- **Optimization Techniques:** By studying Dijkstra's approach, you will develop skills to optimize the selection process of the shortest path in connected graphs.



Evolution Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet		8 Marks
2.	Viva		10 Marks
3.	Simulation		12 Marks
	Total		30 Marks

Teacher Signature|