

DSA Practical\codes\7thPractical.py

```
1 def dijkstra(graph, source):
2     n = len(graph)
3     visited = [False] * n
4     distance = [float('inf')] * n
5     distance[source] = 0
6
7     for _ in range(n):
8         min_dist = float('inf')
9         min_node = -1
10        for i in range(n):
11            if not visited[i] and distance[i] < min_dist:
12                min_dist = distance[i]
13                min_node = i
14
15        if min_node == -1:
16            break
17
18        visited[min_node] = True
19
20        for neighbor in range(n):
21            if graph[min_node][neighbor] != 0 and not visited[neighbor]:
22                new_dist = distance[min_node] + graph[min_node][neighbor]
23                if new_dist < distance[neighbor]:
24                    distance[neighbor] = new_dist
25
26    print(f"\nShortest distances from node {source}:")
27    for i in range(n):
28        if distance[i] == float('inf'):
29            print(f"Node {i}: Unreachable")
30        else:
31            print(f"Node {i}: {distance[i]}")
32
33    graph = [
34        # A B C D E F G H I
35        [ 0, 5, 0, 0, 0, 0, 9,18, 1], # A
36        [ 5, 0, 7, 0, 0, 0, 0, 0, 0], # B
37        [ 0, 7, 0, 8,11, 0, 0, 0, 6], # C
38        [ 0, 0, 8, 0,20, 0, 0, 0, 0], # D
39        [ 0, 0,11,20, 0, 4, 0, 0, 2], # E
40        [ 0, 0, 0, 0, 4, 0, 1, 0, 0], # F
41        [ 9, 0, 0, 0, 0, 0, 1, 0, 2, 3], # G
42        [18, 0, 0, 0, 0, 0, 0, 2, 0, 0], # H
43        [ 1, 0, 6, 0, 2, 0, 3, 0, 0], # I
44    ]
45
46
47 def main():
48     print("Graph has 9 nodes (0 to 8)")
```

```
49     source = int(input("Enter the source node: "))
50     dijkstra(graph, source)
51
52 if __name__ == "__main__":
53     main()
54
55
56
```