5.Given a graph represented by an adjacency matrix, implement Dijkstra's Algorithm to find the shortest path from a given source vertex to all other vertices in the graph. The graph is represented as an adjacency matrix where graph[i][j] denote the weight of the edge from vertex i to vertex j. If there is no edge between vertices i and j, the value is Infinity (or a very large number).

Test Case 1:

Input:

n = 5

graph = [[0, 10, 3, Infinity, Infinity], [Infinity, 0, 1, 2, Infinity], [Infinity, 4, 0, 8, 2],

[Infinity, Infinity, Infinity, 0, 7], [Infinity, Infinity, Infinity, 9, 0]]

source = 0

Output: [0, 7, 3, 9, 5]

Test Case 2:

Input:

n = 4

graph = [[0, 5, Infinity, 10], [Infinity, 0, 3, Infinity], [Infinity, Infinity, 0, 1],

[Infinity, Infinity, Infinity, 0]]

source = 0

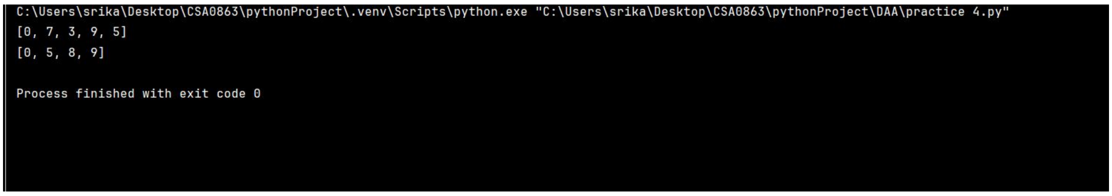
Output: [0, 5, 8, 9]

Program:

```
import heapq

def dijkstra(graph, source):

    n = len(graph)

    distances = [float('inf')] * n

    distances[source] = 0

    min_heap = [(0, source)]

    visited = [False] * n

    while min_heap:

        current_distance, u = heapq.heappop(min_heap)
        if visited[u]:
            continue
        visited[u] = True
        for v in range(n):
            if graph[u][v] != float('inf') and not visited[v]:
                new_distance = current_distance + graph[u][v]
                if new_distance < distances[v]:
                    distances[v] = new_distance
                    heapq.heappush(min_heap, (new_distance, v))
    return distances
n = 5
graph = [
    [0, 10, 3, float('inf'), float('inf')],
    [float('inf'), 0, 1, 2, float('inf')],
    [float('inf'), 4, 0, 8, 2],
    [float('inf'), float('inf'), float('inf'), 0, 7],
    [float('inf'), float('inf'), float('inf'), 9, 0]
]
source = 0
print(dijkstra(graph, source))  # Output: [0, 7, 3, 9, 5]
n = 4
graph = [
    [0, 5, float('inf'), 10],
    [float('inf'), 0, 3, float('inf')],
    [float('inf'), float('inf'), 0, 1],
    [float('inf'), float('inf'), float('inf'), 0]
]
```

```
source = 0

print(dijkstra(graph, source))
```

Output:



```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe "C:\Users\srika\Desktop\CSA0863\pythonProject\DAA\practice 4.py"
[0, 7, 3, 9, 5]
[0, 5, 8, 9]

Process finished with exit code 0
```

Time complexity:

O(n^2logn)