

Name: Ganesh K

USN : 2VX23UE015

Expt No.4(A): Floyd's Algorithm.

```
In [3]: 1 n = 4
2 inf = float('inf')
3 graph = [[0, 3, inf, 7],
4           [8, 0, 2, inf],
5           [5, inf, 0, 1],
6           [2, inf, inf, 0]]
7
8 def floyd_warshall(graph):
9     n = len(graph)
10    dist = []
11    for row in graph:
12        dist.append(row[:])
13
14    for k in range(n):
15        for i in range(n):
16            for j in range(n):
17                dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
18    return dist
19
20 shortest_paths = floyd_warshall(graph)
21
22 print("Shortest paths between all pairs of vertices:")
23 for row in shortest_paths:
24     for dist in row:
25         if dist == inf:
26             print("inf", end=" ")
27         else:
28             print(dist, end=" ")
29
30 print()
31
```

Shortest paths between all pairs of vertices:

```
0 3 5 6
5 0 2 3
3 6 0 1
2 5 7 0
```

```

In [2]: 1 n = int(input("Enter the number of Vertices: "))
2 inf = float('inf')
3 print("Enter the adjacency matrix (Enter 'inf' for infinity):")
4 graph = []
5
6 for i in range(n):
7     row = input().split()
8     graph.append([inf if x == 'inf' else int(x) for x in row])
9
10 def floyd_warshall(graph):
11     n = len(graph)
12     dist = []
13     for row in graph:
14         dist.append(row[:])
15
16     for k in range(n):
17         for i in range(n):
18             for j in range(n):
19                 dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
20     return dist
21
22 shortest_paths = floyd_warshall(graph)
23
24 print("Shortest paths between all pairs of vertices:")
25 for row in shortest_paths:
26     for dist in row:
27         if dist == inf:
28             print("inf", end=" ")
29         else:
30             print(dist, end=" ")
31
32     print()
33

```

```

Enter the number of Vertices: 4
Enter the adjacency matrix (Enter 'inf' for infinity):
0 3 inf 7
8 0 2 inf
5 inf 0 1
2 inf inf 0
Shortest paths between all pairs of vertices:
0 3 5 6
5 0 2 3
3 6 0 1
2 5 7 0

```

In []: 1