

# DESIGN ANALYSIS AND ALGORITHMS

## LAB ASSIGNMENT-6

NAME : PRATHAPANI SATWIKA

REG.NO. : 20BCD7160

**Q) Implementation of all pairs shortest path problem using Dynamic Programming.**

**CODE :**

```
package Lab6;
import java.io.*;
import java.lang.*;
import java.util.*;
    public class allpairshortestpath {
        final static int INF = 99999, V = 4;

        void floydWarshall(int graph[][])
        {
            int dist[][] = new int[V][V];
            int i, j, k;
            for (i = 0; i < V; i++)
                for (j = 0; j < V; j++)
                    dist[i][j] = graph[i][j];
            for (k = 0; k < V; k++) {
                for (i = 0; i < V; i++) {
                    for (j = 0; j < V; j++) {
                        if (dist[i][k] + dist[k][j]
                            < dist[i][j])
                            dist[i][j]
                                = dist[i][k] + dist[k][j];
                    }
                }
            }
            printSolution(dist);
        }

        void printSolution(int dist[][])
        {
            System.out.println("The following matrix shows the shortest " +
                "distances between every pair of vertices");
            for (int i = 0; i < V; ++i) {
                for (int j = 0; j < V; ++j) {
                    if (dist[i][j] == INF)
                        System.out.print("INF ");
                    else
                        System.out.print(dist[i][j] + " ");
                }
                System.out.println();
            }
        }
    }
```

```

    }
    public static void main(String[] args)
    {
        int graph[][] = { { 0, 4, INF, 8 },
                           { INF, 0, 6, INF },
                           { INF, INF, 0, 3 },
                           { INF, INF, INF, 0 } };
        allpairshortestpath a = new allpairshortestpath();
        a.floydWarshall(graph);
    }
}

1 package Lab6;
2 import java.io.*;
3 import java.lang.*;
4 import java.util.*;
5 public class allpairshortestpath {
6     final static int INF = 99999, V = 4;
7
8     void floydWarshall(int graph[][])
9     {
10         int dist[][] = new int[V][V];
11         int i, j, k;
12         for (i = 0; i < V; i++)
13             for (j = 0; j < V; j++)
14                 dist[i][j] = graph[i][j];
15         for (k = 0; k < V; k++) {
16             for (i = 0; i < V; i++) {
17                 for (j = 0; j < V; j++) {
18                     if (dist[i][k] + dist[k][j]
19                         < dist[i][j])
20                         dist[i][j]
21                             = dist[i][k] + dist[k][j];
22                 }
23             }
24         }
25         printSolution(dist);
26     }
27
28     void printSolution(int dist[][])
29     {
30         System.out.println("The following matrix shows the shortest " + "distances between every pair of vertices");
31         for (int i = 0; i < V; ++i) {
32             for (int j = 0; j < V; ++j) {
33                 if (dist[i][j] == INF)
34                     System.out.print("INF ");
35                 else
36                     System.out.print(dist[i][j] + " ");
37             }
38             System.out.println();
39         }
40     }
41
42     public static void main(String[] args)
43     {
44         int graph[][] = { { 0, 4, INF, 8 },
45                           { INF, 0, 6, INF },
46                           { INF, INF, 0, 3 },
47                           { INF, INF, INF, 0 } };
48         allpairshortestpath a = new allpairshortestpath();
49         a.floydWarshall(graph);
50     }
51 }

```

## OUTPUT :

```
<terminated> allpairshortestpath [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe (Oct 15, 2022, 10:31:34 AM -  
The following matrix shows the shortest distances between every pair of vertices  
0 4 10 8  
INF 0 6 9  
INF INF 0 3  
INF INF INF 0
```