Comps-A Batch-C

Mark Lopes

9913

Flyod Warshall :-

```c
#include <stdio.h>
#include <limits.h>

#define INF INT_MAX
#define V 4

void printSolution(int dist[][V])
{
    printf("Shortest distances between every pair of vertices:\n");
    for (int i = 0; i < V; i++)
    {
        for (int j = 0; j < V; j++)
        {
            if (dist[i][j] == INF)
                printf("INF\t");
            else
                printf("%d\t", dist[i][j]);
        }
        printf("\n");
    }
}

void floydWarshall(int graph[][V])
{
    int dist[V][V];

    // Initialize distance matrix
    for (int i = 0; i < V; i++)
    {
        for (int j = 0; j < V; j++)
        {
            dist[i][j] = graph[i][j];
        }
    }

    // Applying Floyd-Warshall algorithm
    for (int k = 0; k < V; k++)
    {
```

```
        for (int i = 0; i < V; i++)
        {
            for (int j = 0; j < V; j++)
            {
                if (dist[i][k] != INF && dist[k][j] != INF && dist[i][k] +
dist[k][j] < dist[i][j])
                {
                    dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
    }

    printSolution(dist);
}

int main()
{
    int graph[V][V] = {
        {0, INF, -2, INF},
        {4, 0, 3, INF},
        {INF, INF, 0, 2},
        {INF, -1, INF, 0}};

    floydWarshall(graph);
    return 0;
}
```

```
Shortest distances between every pair of vertices:
0       -1      -2      0
4       0       2       4
5       1       0       2
3       -1      1       0
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\AoA>
```