	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of** **Artificial Intelligence and Data Science**	
Student Name: Mohammad Faiz Nishat Parvej Saiyad		
Class: TY	Division: A	Roll No:371034
Semester: V	Academic Year:2022-23	
Subject Name & Code: Design and Analysis of Algorithms		
Title of Assignment: **Implement All Pair Shortest paths problem using Floyd's Algorithm.**		
Date of Performance:	Date of Submission:	

Aim:

Implement All Pair Shortest paths problem using Floyd's Algorithm.

Problem Statement:

Implement All Pair Shortest paths problem using Floyd's Algorithm.

Software Requirements:

Text Editor: VSCode, Neovim, etc

Environment: Python 3.10

Terminal Emulator

Background Information:

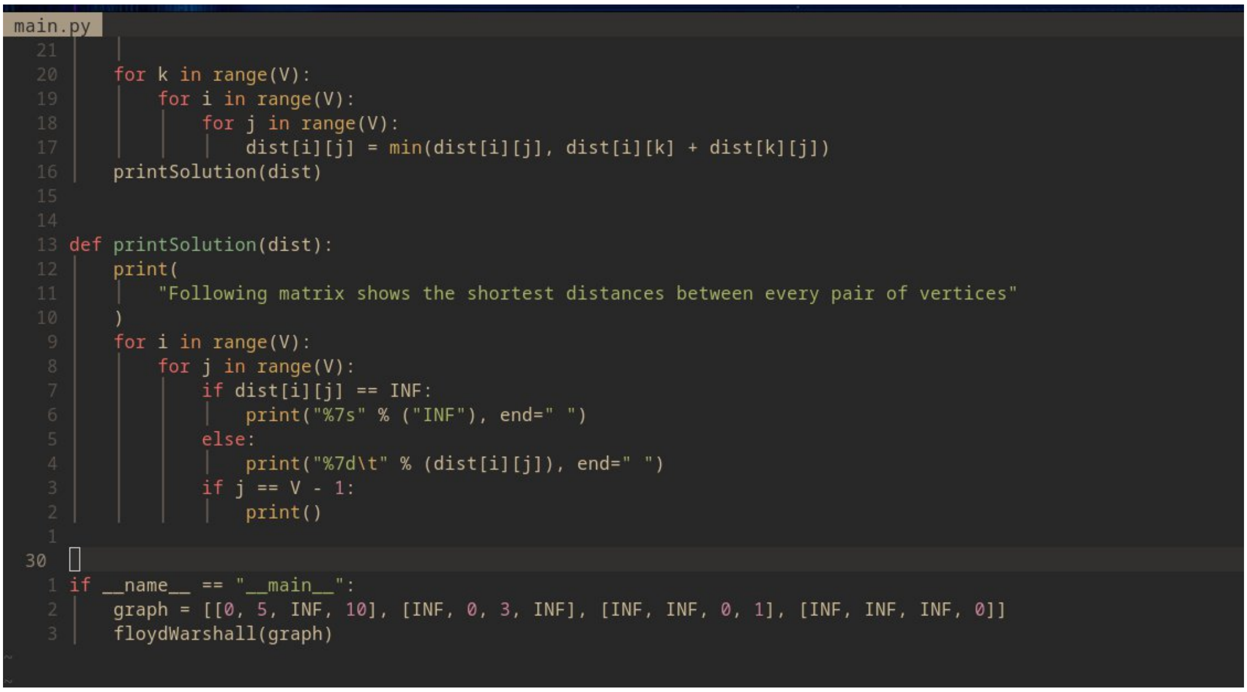
The Floyd Warshall Algorithm:

The Floyd Warshall Algorithm is for solving all pairs shortest path problems. The problem is to find the shortest distances between every pair of vertices in a given edge-weighted directed Graph.

Algorithm:

- Initialize the solution matrix same as the input graph matrix as a first step.
- Then update the solution matrix by considering all vertices as an intermediate vertex.
- The idea is to one by one pick all vertices and updates all shortest paths which include the picked vertex as an intermediate vertex in the shortest path.
- When we pick vertex number k as an intermediate vertex, we already have considered vertices {0, 1, 2, .. k-1} as intermediate vertices.
- For every pair (i, j) of the source and destination vertices respectively, there are two possible cases.
- k is not an intermediate vertex in shortest path from i to j. We keep the value of dist[i][j] as it is.
- k is an intermediate vertex in shortest path from i to j. We update the value of dist[i][j] as dist[i][k] + dist[k][j] if dist[i][j] > dist[i][k] + dist[k][j]

Code:

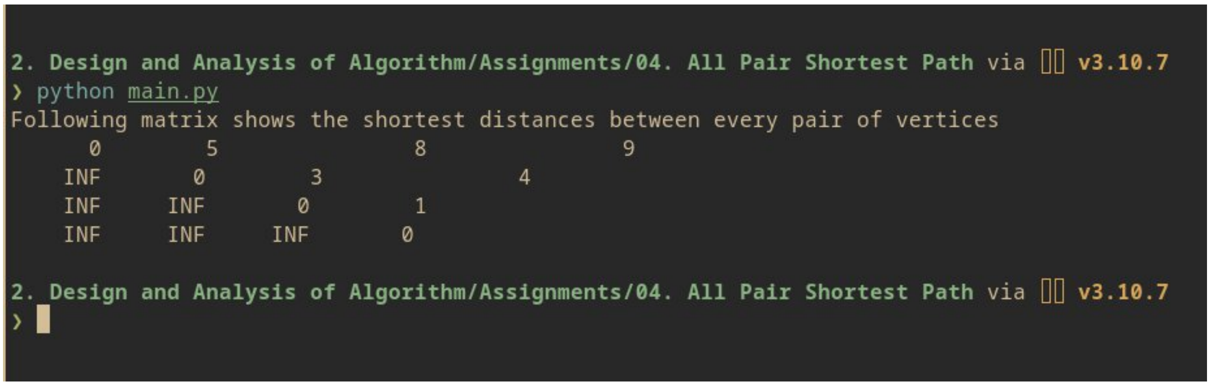


```
    for k in range(V):
        for i in range(V):
            for j in range(V):
                dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
    printSolution(dist)


def printSolution(dist):
    print(
        "Following matrix shows the shortest distances between every pair of vertices"
    )
    for i in range(V):
        for j in range(V):
            if dist[i][j] == INF:
                print("%7s" % ("INF"), end=" ")
            else:
                print("%7d\t" % (dist[i][j]), end=" ")
            if j == V - 1:
                print()

if __name__ == "__main__":
    graph = [[0, 5, INF, 10], [INF, 0, 3, INF], [INF, INF, 0, 1], [INF, INF, INF, 0]]
    floydWarshall(graph)
```

Output:



```
2. Design and Analysis of Algorithm/Assignments/04. All Pair Shortest Path via  v3.10.7
❯ python main.py
Following matrix shows the shortest distances between every pair of vertices
      0        5               8               9
    INF       0        3               4
    INF     INF       0        1
    INF     INF     INF       0

2. Design and Analysis of Algorithm/Assignments/04. All Pair Shortest Path via  v3.10.7
❯ 
```

Conclusion:

Implemented All Pairs Shortest Path Problem using Floyd Warshall Algorithm.