K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

Batch:C2	**Roll No.:16010122257**
Experiment No._7_	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Title: Implementation of All Pair Shortest Path using Dynamic Programming

Objective To learn the All-Pair Shortest Path using Floyd-Warshall's algorithm

CO to be achieved:

CO 2 Describe various algorithm design strategies to solve different problems and analyse Complexity.

Books/ Journals/ Websites referred:

1. **Ellis horowitz, Sarataj Sahni, S.Rajsekaran," Fundamentals of computer algorithm", University Press**
2. **T.H.Cormen ,C.E.Leiserson,R.L.Rivest and C.Stein," Introduction to algortihtms",2nd Edition ,MIT press/McGraw Hill,2001**
3. **http://users.cecs.anu.edu.au/~Alistair.Rendell/Teaching/apac_comp3600/module 4/all_pairs_shortest_paths.xhtml**
4. **https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/**
5. **http://www.cs.bilkent.edu.tr/~atat/502/AllPairsSP.ppt**

Theory:

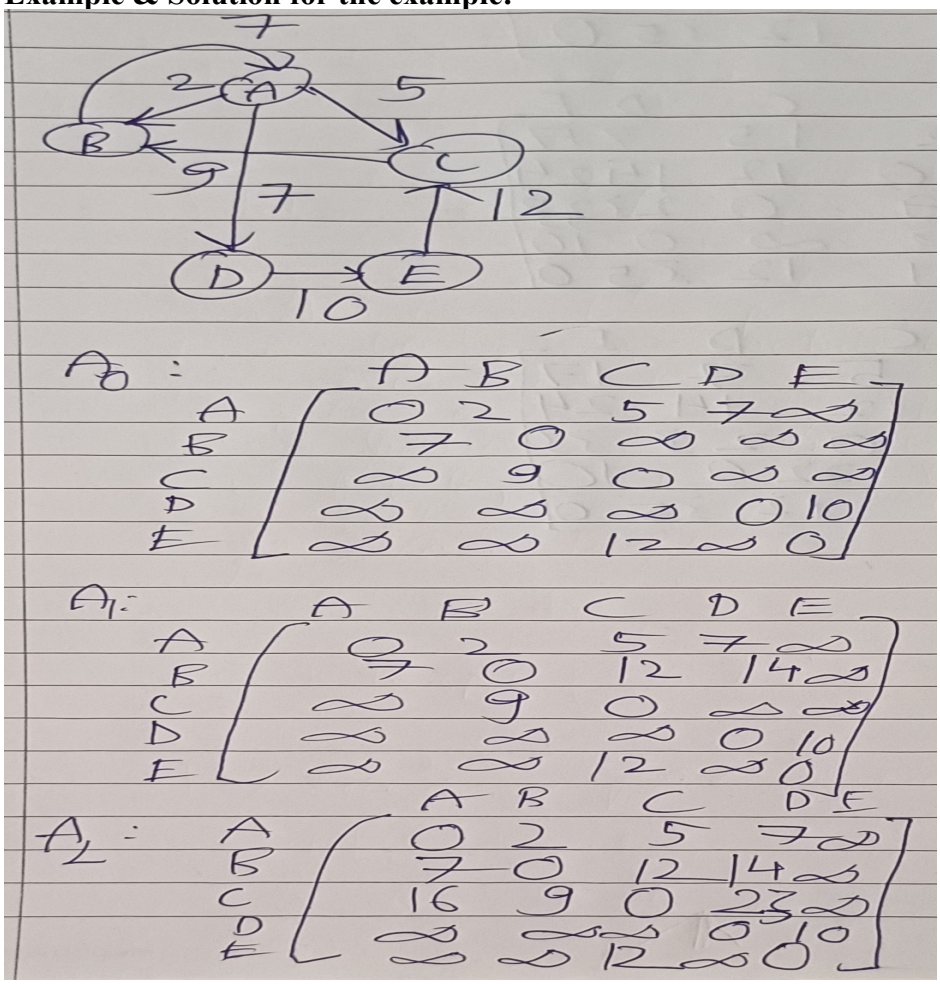
It aims to figure out the shortest path from each vertex v to every other u.

1. In all pair shortest path, when a weighted graph is represented by its weight matrix W then objective is to find the distance between every pair of nodes.
2. Apply dynamic programming to solve the all pairs shortest path.
3. In all pair shortest path algorithm, we first decomposed the given problem into sub problems.
4. In this principle of optimally is used for solving the problem.
5. It means any sub path of shortest path is a shortest path between the end nodes.

Algorithm:

```
Algorithm All_pair(W, A)
{
For i = 1 to n do
For j = 1 to n do
A [i , j] = W [i , j]
For k = 1 to n do
        {
        For i = 1 to n do
                {
                For j = 1 to n do
                {
                A [i , j] = min(A [i, j], A [i, k] + A [k, j])
                }
                }}}
```

Example & Solution for the example:



A_3:

	A	B	C	D	E
A	0	2	15	7	∞
B	7	0	12	14	∞
C	16	9	0	23	∞
D	∞	∞	∞	0	10
E	28	21	12	35	0

A_4:

	A	B	C	D	E
A	0	2	15	7	17
B	7	0	12	14	24
C	16	9	0	23	33
D	∞	∞	∞	0	10
E	28	21	12	35	0

A_5:

	A	B	C	D	E
A	0	2	5	7	17
B	7	0	12	14	24
C	16	9	0	23	33
D	38	31	22	0	10
E	28	21	12	35	0

Analysis of algorithm:

- It uses 3 nested loops. Innermost loop has only 1 statement. T.C. = $\theta(1)$.
- Running time of the algo is computed as:

$$T(V) = \sum_{k=1}^{V} \sum_{i=1}^{V} \sum_{j=1}^{V} \theta(1)$$

$$= \sum_{k=1}^{V} \sum_{i=1}^{V} V = \sum_{k=1}^{V} V^2 = \theta(V^3)$$

- Thus, Floyd's algo. runs in cubic time.

CODE:

main.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

#define v 5
#define INF 99999

void floydWarshallAlgo(int graph[v][v]) {
    int distance[v][v];

    for(int i=0; i<v; ++i)
        for(int j=0; j<v; ++j)
            distance[i][j] = graph[i][j];


    for(int k=0; k<v; ++k)
        for(int i=0; i<v; ++i)
            for(int j=0; j<v; ++j)
            {
                if(distance[i][k] == INF || distance[k][j] == INF)
                    continue;
                else if(distance[i][k] + distance[k][j] < distance[i][j])
                    distance[i][j] = distance[i][k] + distance[k][j];
            }


    for(int i=0; i<v; ++i)
        if(distance[i][i] < 0) {
            cout << "Oh!Negative edge weight cycle is present\n";
            return;
        }


    cout << "Following matrix shows the shortest distances between every pair of vertices \n";
    for(int i=0; i<v; ++i) {
        for(int j=0; j<v; ++j) {
            if(distance[i][j] == INF)
                cout << "Infinity" << " ";
            else
                cout << distance[i][j] << " ";
        }
        cout << endl;
    }
}

int main() {
    int graph[v][v] = { {0, 2, 5, 7, INF},
                        {7, 0, INF, INF, INF},
                        {INF, 9, 0, INF, INF},
                        {INF, INF, INF, 0, 10},
                        {INF, INF, 12, INF, 0} };

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

OUTPUT:

```
input
The following matrix shows the shortest distances between every pair of vertices
0 2 5 7 17
7 0 12 14 24
16 9 0 23 33
38 31 22 0 10
28 21 12 35 0


...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION: Thus,from this experiment we've learnt and understood the working of all pair shortest path algorithm using the Floyd-Warshall algorithm.We solved an example graph of it in class and implemented it in C++.It's an algorithm which is used to find the shortest path from one node to all other nodes in any graph. It takes O(v^3) time complexity and O(v^2) space complexity.