

| **Title: Implementation of All Pair Shortest Path using Dynamic Programming** |
| --- |

**Objective** To learn the All-Pair Shortest Path using Floyd-Warshall’salgorithm

**CO to be achieved:**

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

**Books/ Journals/ Websites referred:**

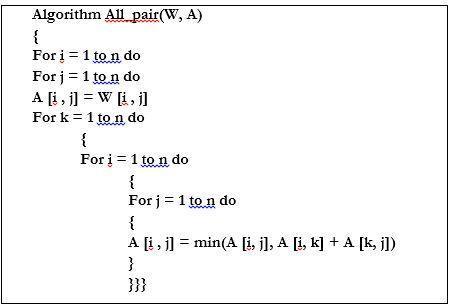
1. **Ellis horowitz, Sarataj Sahni, S.Rajsekaran,” Fundamentals of computer algorithm”, University Press**
2. **T.H.Cormen ,C.E.Leiserson,R.L.Rivest and C.Stein,” Introduction to algortihtms”,2nd Edition ,MIT press/McGraw Hill,2001**
3. **http://users.cecs.anu.edu.au/~Alistair.Rendell/Teaching/apac\_comp3600/module4/all\_pairs\_shortest\_paths.xhtml**
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:**



**Example :**

**CODE:**

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin >> n;

vector<vector<int>> v(n, vector<int> (n)), path(n, vector<int> (n));

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

cin >> v[i][j];

}

}

for(int k = 0; k < n; k++)

{

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

// v[i][j] = min(v[i][j], v[i][k] + v[k][j]);

if(v[i][k] + v[k][j] < v[i][j])

{

path[i][j] = k + 1;

v[i][j] = v[i][k] + v[k][j];

}

}

}

}

cout << "Dist " << endl << endl;

for(auto &i: v){

for(auto &j: i)cout << j << " ";

cout << endl;

}

cout << endl << "Path " << endl << endl;

for(auto &i: path){

for(auto &j: i)cout << j << " ";

cout << endl;

}

cout << endl << "Displaying paths" << endl;

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

if(v[i][j] != 999)

{

cout << "Intermediate nodes from: " << i + 1 << " " << j + 1<< ": ";

int x = path[i][j];

while(x != 0)

{

cout << x << " ";

x = path[x-1][j];

}

cout << endl;

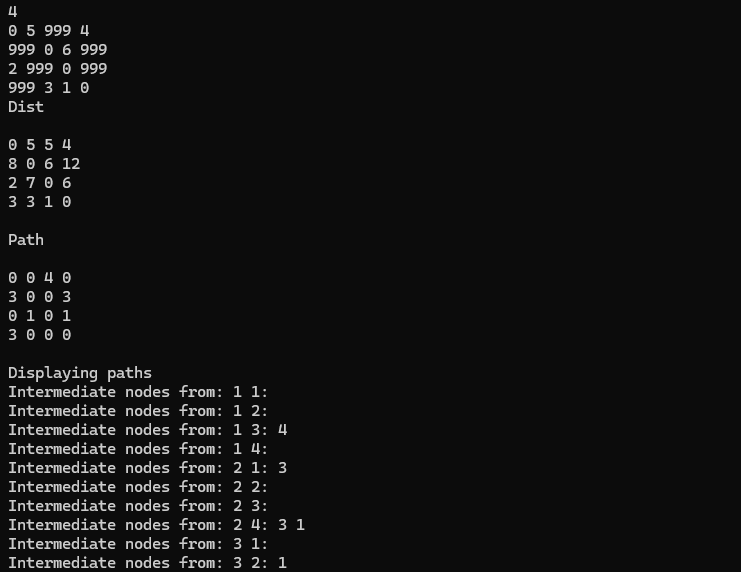
}

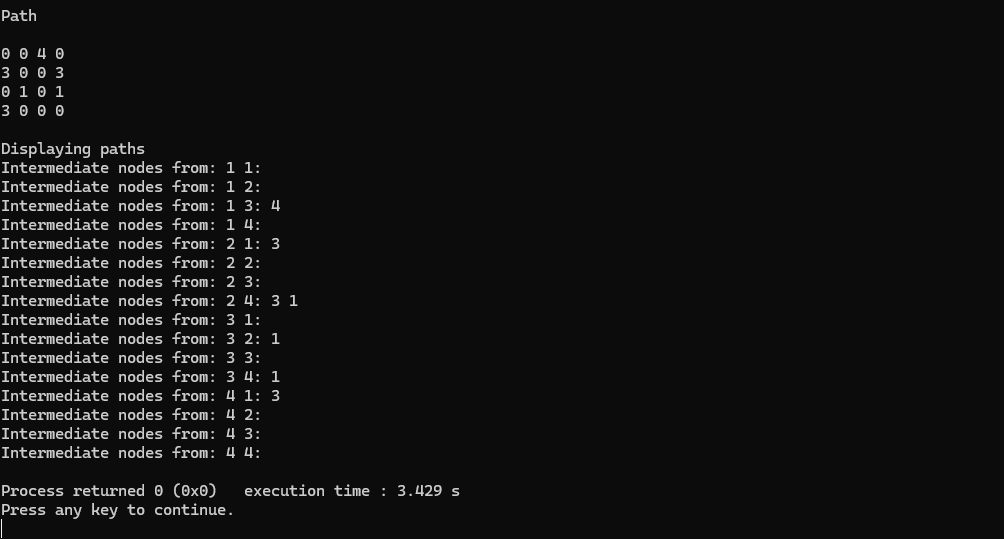
}

}

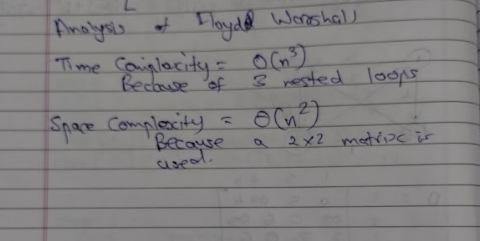
}

**OUTPUT:**





**Analysis of algorithm:**

****

**CONCLUSION:** The implementation of the Floyd-Warshall algorithm for All-Pair Shortest Path provides insight into dynamic programming techniques. Understanding its design strategies aids in analyzing complexities of diverse problem-solving approaches.