**Batch: B3 Roll No.: 121**

**Experiment No.\_6\_**

**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’salgorithm

**CO to be achieved:**

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

**Books/ Journals/ Websites referred:**

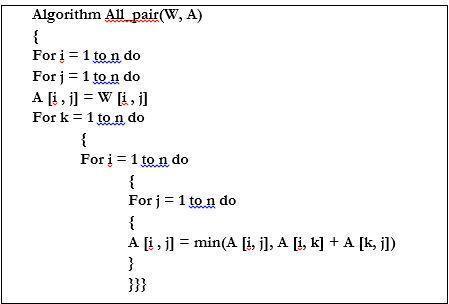
1. **Ellis horowitz, Sarataj Sahni, S.Rajasekaran,” Fundamentals of computer algorithm”, University Press**
2. **T.H.Cormen ,C.E.Leiserson,R.L.Rivest and C.Stein,” Introduction to algorithms”,2nd Edition ,MIT press/McGraw Hill,2001**
3. **http://users.cecs.anu.edu.au/~Alistair.Rendell/Teaching/apac\_comp3600/module4/all\_pairs\_shortest\_paths.xhtml**
4. **https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/**
5. **http://www.cs.bilkent.edu.tr/~atat/502/AllPairsSP.ppt**
6. **https://www.gyaanibuddy.com/assignments/assignment-detail/dynamic-programming-all-pair-shortest-path/**
7. **https://www.codingninjas.com/codestudio/library/all-pair-shortest-path**

**Theory:**

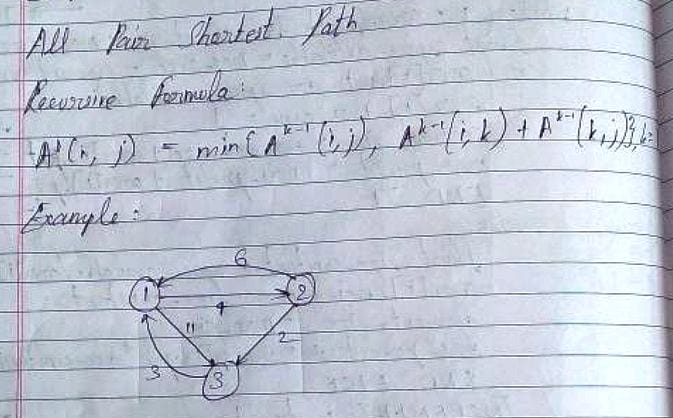
It aims to figure out the shortest path from each vertex v to every other u.

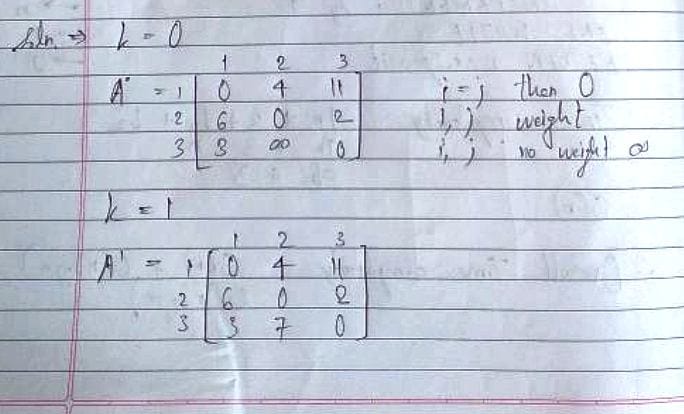
1. In all pairs shortest path, when a weighted graph is represented by its weight matrix W then the objective is to find the distance between every pair of nodes.
2. Apply dynamic programming to solve the all pairs shortest path.
3. In the all pair shortest path algorithm, we first decomposed the given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

**Algorithm:**

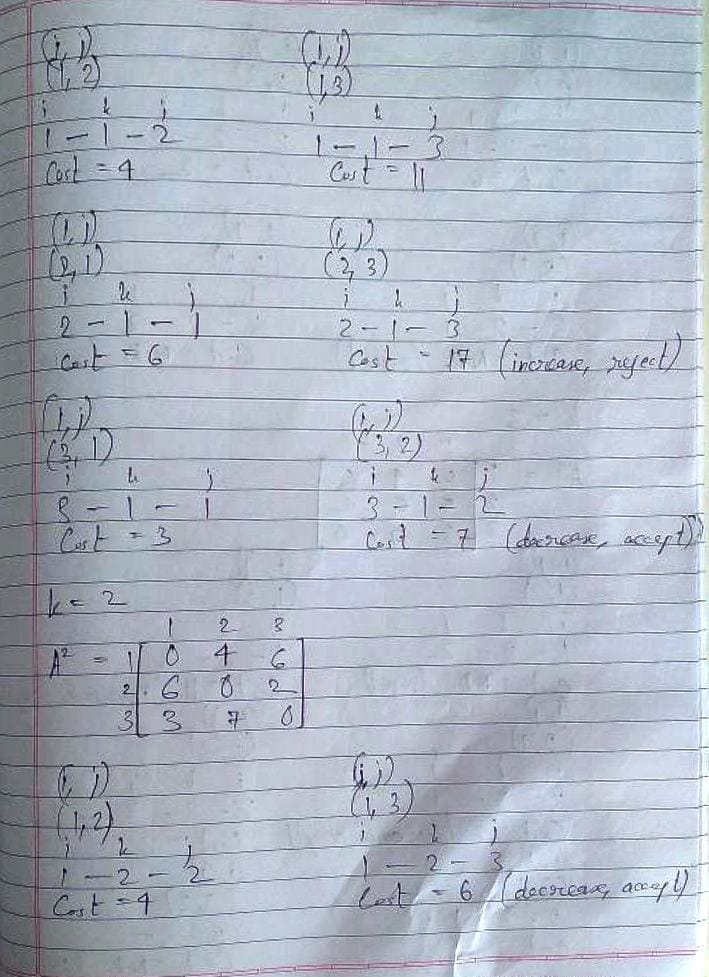


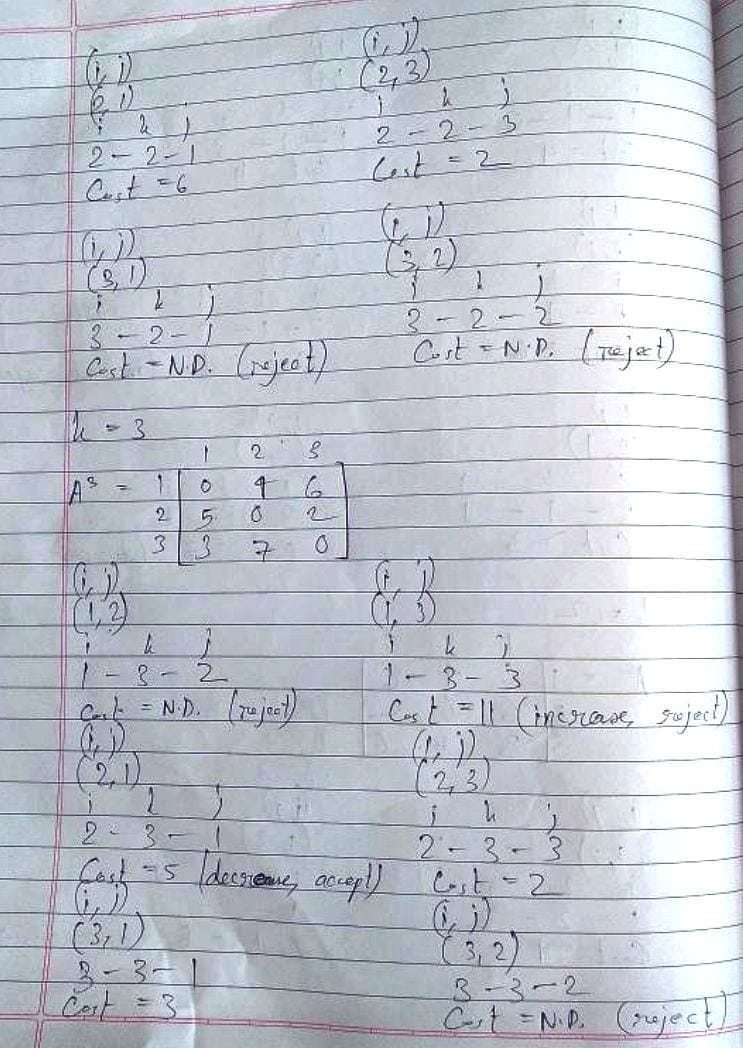
**Example:**

****



**Solution for the example:**





**Analysis of 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 … … … … … … … … … … … … … 🡪 n + 1

{

For i = 1 to n do … … … … … … … … … … … … 🡪 (n + 1) n

{

For j = 1 to n do… … … … … … … … … … 🡪(n + 1) n n

{

A[i, j] = min{A[i, j], A[i,k] + A[k,j]}

}

}

}

}

Total time complexity: n3 + n2 + n2 + n + n + 1

= O(n3)

The space complexity is O(v2) where, 'v' is the number of vertices in the graph. This is because the details for all the vertices and the distance between any pair of vertices is stored in a v-by-v matrix.

**CONCLUSION:**

Thus, in this experiment, the concept of All Pair Shortest Paths was learnt and implemented. Implementing this algorithm was quite easy, with three simple and straight for loops. So, it can be a popular choice among developers when there is a need to find the shortest distance between any pair of vertices. However, it does not have a good time complexity which means that it will take a long time for larger graphs having large number of vertices (in the order of 1000s or more) with complicated interconnectivity.