|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Analysis of Algorithms** | **Semester:** | **IV** |
| **Date of Performance:** | **27 / 03 / 2024** | **Batch No:** | **A – 2** |
| **Faculty Name:** | **Dr. Aarti Phadke** | **Roll No.:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade / Marks:** | **\_\_\_ / 25** |

**Experiment No.: 7**

**Title: All Pair Shortest Path**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Implementation of All Pair Shortest Path using Dynamic Programming.  **Objective:** To learn the All-Pair Shortest Path using Floyd-Warshall’s algorithm. |

|  |
| --- |
| **COs to be achieved:** |
| **CO2:** Describe various algorithm design strategies to solve different problems. |

|  |
| --- |
| **Apparatus / Software Tools Used:** |
| 1. VS Code 2. Microsoft Excel |

|  |
| --- |
| **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 sub path of shortest path is a shortest path between the end nodes. |

|  |
| --- |
| **Stepwise-Procedure / Algorithm:** |
| Image result for all pair shortest path algorithm |

|  |
| --- |
| **Upload the code / Output:** |
| Code:  INF = float('inf')  def all\_pair\_shortest\_path(W):      n = len(W)      A = [[0] \* n for \_ in range(n)]        for i in range(n):          for j in range(n):              A[i][j] = W[i][j]        for k in range(n):          for i in range(n):              for j in range(n):                  A[i][j] = min(A[i][j], A[i][k] + A[k][j])        return A  def input\_matrix(n):      matrix = []      print("enter elements of the matrix ('inf' for infinity): ")      for i in range(n):          row = input(f"enter elements of row {i+1}: ").split()          row = [INF if x == 'inf' else int(x) for x in row]          matrix.append(row)      return matrix  print("\nAll Pair Shortest Path\n")  n = int(input("enter the size of matrix: "))  W = input\_matrix(n)  A = all\_pair\_shortest\_path(W)  print("\nresulting matrix A (Shortest Paths): ")  for row in A:      print(row)  Output: |

|  |
| --- |
| **Post Lab Subjective / Objective Type Questions:** |
| **For the graph (which was implement during the practical), apply Floyd Warshall’s Algorithm to compute all pairs shortest distance.**  **Show all steps.**  Handwritten Solution: |

|  |
| --- |
| **Conclusion:** |
| Through dynamic programming, our experiment successfully demonstrated the efficient computation of all pair shortest paths, providing valuable insights into algorithmic analysis and optimization. |

**Signature of faculty in-charge with Date:**