UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Worksheet 5(b)

Student Name: Rahul Saxena **UID:** 24MCI10204

Branch: MCA(AI&ML) **Section/Group:** 3-B

Semester: 1st semester **Date of Performance:** 11/10/2024

Subject Name: Design and analysis of Algorithm Lab **Subject Code:** 24CAP-612

AIM:

Compute the transitive closure of a given directed graph using Warshall's algorithm

Task To be Done:

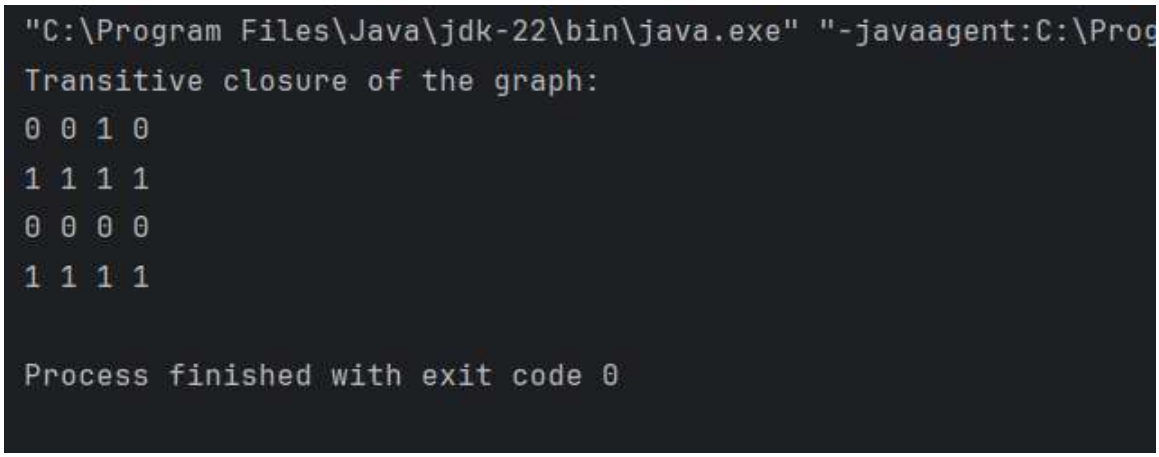
- **Understand the Transitive Closure Concept:** Learn what it means to find the transitive closure of a directed graph, which involves determining if there is a path between each pair of vertices.
- **Implement Warshall's Algorithm:** Use Warshall's algorithm to calculate the transitive closure of a given directed graph. Warshall's algorithm is a classic dynamic programming approach to solve this problem.
- **Develop a Program:** Write a program that takes a directed graph as an adjacency matrix and computes its transitive closure using Warshall's algorithm.

Source Code:

```
public class WarshallAlgorithm {
   public static void warshallAlgorithm(int[][] graph) {
      int n = graph.length;
      for (int k = 0; k < n; k++) {
         for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
               graph[i][j] = (graph[i][j] != 0) || (graph[i][k] != 0 && graph[k][j] != 0) ? 1 : 0;
            }
         }
      }
      System.out.println("Transitive closure of the graph:");
      for (int[] ints : graph) {
         for (int j = 0; j < n; j++) {
            System.out.print(ints[j] + " ");
         }
         System.out.println();
      }
   }
```

```
    public static void main(String[] args) {
        int[][] graph = {
                {0, 0, 1, 0},
                {1, 0, 0, 1},
                {0, 0, 0, 0},
                {0, 1, 0, 0}
        };
        warshallAlgorithm(graph);
    }
}
```

Output:



Learning Outcome:

- Gain a solid understanding of graph theory concepts, specifically transitive closure and how it relates to connectivity in directed graphs.
- Learn to implement and apply Warshall's algorithm for solving real-world graph problems.
- Improve skills in Java programming, especially in working with matrices and nested loops.
- Develop analytical skills in understanding the efficiency and application of dynamic programming to graph problems.