F) Write a program to solve a  Travelling Salesman Problem using dynamic programming or branch and

bound strategy.

Aarti Ambekar

RollNo:20121056

import java.util.Arrays;

public class TSPDynamicProgramming {

    public static int tsp(int[][] graph) {

        int n = graph.length;

        int[][] dp = new int[1 << n][n];

        for (int[] row : dp) {

            Arrays.fill(row, Integer.MAX\_VALUE);

        }

        dp[1][0] = 0;

        for (int mask = 1; mask < (1 << n); mask++) {

            for (int u = 0; u < n; u++) {

                if ((mask & (1 << u)) != 0) {

                    for (int v = 0; v < n; v++) {

                        if (u != v && (mask & (1 << v)) != 0) {

                            dp[mask][v] = Math.min(dp[mask][v], dp[mask ^ (1 << v)][u] + graph[u][v]);

                        }

                    }

                }

            }

        }

        int finalMask = (1 << n) - 1;

        int minCost = Integer.MAX\_VALUE;

        for (int v = 1; v < n; v++) {

            if (graph[v][0] != 0) {

                minCost = Math.min(minCost, dp[finalMask][v] + graph[v][0]);

            }

        }

        return minCost;

    }

    public static void main(String[] args) {

        int[][] graph = {

            {0, 29, 20, 21},

            {29, 0, 15, 12},

            {20, 15, 0, 23},

            {21, 12, 23, 0}

        };

        int minCost = tsp(graph);

        System.out.println("Minimum cost for TSP: " + minCost);

    }

}

Ouptput:

Minimum cost for TSP: 67