Name: Aditya.Rajesh.Wanwade

Roll no: 27

Branch: Cyber Security

DAA Practical no:5

Code of Travelling Salesman Problem:

```
import java.util.*;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.LinkedHashSet;
public class TSP {
    public static int TSP(int[][] DP, int mark, int position, int number,
int[][] adj, List<Integer> path) {
        int completed_visit = (1 << number) - 1;
        if (mark == completed_visit) {
            path.add(0);
            return adj[position][0];
        }

        if (DP[mark][position] != -1) {
            return DP[mark][position];
        }

        int answer = Integer.MAX_VALUE;
        int nextCity = -1;

        for (int city = 0; city < number; city++) {
            if ((mark & (1 << city)) == 0) {
                int newAnswer = adj[position][city] + TSP(DP, mark | (1 <<
city), city, number, adj, path);
                if (newAnswer < answer) {
                    answer = newAnswer;
                    nextCity = city;
                }
            }
        }

        DP[mark][position] = answer;

        if (nextCity != -1) {
            path.add(nextCity);
        }
```

```
        return answer;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of locations:");
        int number = sc.nextInt();

        System.out.println("Enter the start location (0-indexed):");
        int s = sc.nextInt();

        int[][] adj = new int[number][number];
        System.out.println("Enter the elements of the Adjacency matrix:");
        for (int i = 0; i < number; i++) {
            for (int j = 0; j < number; j++) {
                adj[i][j] = sc.nextInt();
            }
        }

        int[][] DP = new int[1 << number][number];
        for (int i = 0; i < (1 << number); i++) {
            for (int j = 0; j < number; j++) {
                DP[i][j] = -1;
            }
        }
        DP[(1 << number) - 1][0] = 0;

        List<Integer> path = new ArrayList<>();
        int minCost = TSP(DP, 1 << s, s, number, adj, path);

        System.out.println("Minimum cost: " + minCost);
    Set<Integer> hashSet = new LinkedHashSet<Integer>(path);
        System.out.print("Optimal Path: ");
        for (int city : hashSet) {
            System.out.print((char)(city+65) + " ");
        }
        System.out.println((char)(s+65));
        sc.close();
    }
}
```

Output:


DAA cd "e:\DAA\" ; if ($?) { javac TSP.java } ; if ($?) { java TSP }
Enter the number of locations:
4
Enter the start location (0-indexed):
0
Enter the elements of the Adjacency matrix:
0 16 11 6
8 0 13 16
4 7 0 9
5 12 2 0
Minimum cost: 23
Optimal Path: A D C B A
Asus DAA main ≡ ?7
in pwsh at 01:24:28