

ASSIGNMENT- TRAVELLING SALESMAN PROBLEM USING DFS

A1

PRAJAKTA DEOKULE

3330

```
import java.util.*;
public class TSSP {

    public static void main(String args[]) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the no of cities:");
        int n=sc.nextInt();

        int map[][]=new int[n][n];
        System.out.println("Enter the distance between the cities:");
        for(int i=0;i<n;i++) {
            for(int j=0;j<n;j++) {

                map[i][j]=sc.nextInt();

            }
        }

        System.out.println("Enter the start city:");
        int start=sc.nextInt();

        dfs(n,start-1,map);

    }

    public static void dfs(int n,int start,int graph[][]) {

        Stack<Integer> st=new Stack<Integer>();
        ArrayList<Integer> path=new ArrayList<Integer>();

        boolean visited[]=new boolean[n];
        Arrays.fill(visited, false);

        int count=0,min=0,minCost=0,j=start,ind=0;
        visited[j]=true;
        while(count<n) {

            st.push(j);
            min=Integer.MAX_VALUE;

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

                //city not yet visited
                if(graph[j][i]<min && graph[j][i]>0 && visited[i]==false) {

                    //find the city with the minimum distance from the current
                    city
                    min=graph[j][i];
                    ind=i;

                }
            }
            //close for
            visited[ind]=true;
        }
    }
}
```

```

        minCost=minCost+graph[j][ind];
        j=ind;
        count++;
    }//close while loop

    System.out.println("The path with shortest distance is: ");
    while(st.isEmpty()==false) {
        path.add(st.pop());
    }

    for(int i=path.size()-1;i>=0;i--){
        System.out.print(path.get(i)+1+"->"); //display path
    }

    System.out.println(start+1);
    minCost = minCost + graph[path.get(0)][start];
    System.out.println("\nThe distance travelled is: "+minCost);
}
}

```

```

Enter the no of cities:
4
Enter the distance between the cities:
0 10 15 20
10 0 35 25
15 35 0 30
20 25 30 0
Enter the start city:
1
The path with shortest distance is:
1->2->4->3->1

The distance travelled is: 80

```