

8. Find Minimum Cost Spanning Tree of a given connected undirected graph using **Kruskal's algorithm. Use Union-Find algorithms in your program.**

```
import java.util.*;
public class KruskalDemo
{
    static int parent[];
    static int cost[][];
    public static void main(String [] args)
    {
        int i,j,n,min,ne=1;
        int u=0,v=0,a=0,b=0,mincost=0;

        /* Scanner class to read values from console */
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the number of vertices/nodes in the graph");
        n = in.nextInt();

        cost = new int[n+1][n+1];
        parent= new int[n+1];

        System.out.println("Enter the Cost/Weight matrix");
        for(i=1;i<=n;i++)
        {
            parent[i]=0;
            for(j=1;j<=n;j++)
            {
                cost[i][j] = in.nextInt();
                if(cost[i][j]==0)
                    cost[i][j]=999;
            }
        }

        System.out.println("The edges of Minimum spanning tree are:");
        while(ne<n)
        {
            min = 999;
            for(i=1;i<=n;i++)
            {
                for(j=1;j<=n;j++)
                {
                    if(cost[i][j]<min)
                    {
                        min = cost[i][j];

```

```
        a = u = i;
        b = v = j;
    }
}
}
u = findParent(u);
v = findParent(v);

if(union(u,v)==1)
{
    System.out.println(ne++ + " Edge Selected (" + a + " --- " + b + ") Cost=" + min);
    mincost += min;
}
cost[a][b] = cost[b][a] = 999;
}
System.out.println("Minimum cost " + mincost);
}

/* To find the parent of particular vertex */
public static int findParent(int i)
{
    while(parent[i]!=0)
    {
        i = parent[i];
    }
    return i;
}

/* To determine the cycle in the graph formed by selection of particular edge */
public static int union(int i,int j)
{
    if(i!=j)
    {
        parent[j]=i;
        return 1;
    }
    return 0;
}
}
```