PROGRAM 8

**8)**

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

package labprograms;

import java.util.Scanner;

public class p8 {

static int[] parent=new int[50];

static int[][] cost=new int[50][50];

static int a,b,i,j,u,v,n,min,no\_of\_edges=1,mincost=0;

static int find(int w)

{

while(parent[w]!=0)

w=parent[w];

return w;

}

static void union()

{

if(u!=v)

{

no\_of\_edges++;

System.out.println(no\_of\_edges-1+":Edge("+a+","+b+")="+min);

mincost+=min;

parent[v]=u;

}

cost[a][b]=cost[b][a]=1000;

}

public static void main(String[] args) {

Scanner read=new Scanner(System.in);

System.out.print("Enter the number of vertices : ");

n=read.nextInt();

System.out.println("Enter the cost adjacency matrix,1000 for no direct path : ");

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

{

cost[i][j]=read.nextInt();

if(cost[i][j]==0)

cost[i][j]=1000;

}

System.out.println("The spanning tree has the following edges : ");

while(no\_of\_edges<n)

{

for(i=1,min=1000;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=find(u);

v=find(v);

union();

}

System.out.println("Minimum Cost ="+mincost);

read.close();

}

}