Week 8: **Write a java program that implements Kruskal’s algorithm to generate minimum cost spanning tree.**

import java.util.\*;

public class Kruskal

{

static int p[];

static int n,m,mincost,i,j;

public static void main(String args[]) throws Exception

{

Scanner s=new Scanner(System.in);

System.out.println("enter the number of vertices in the graph\n");

n=s.nextInt();

System.out.println("enter edges in the graph\n");

m=s.nextInt();

p=new int[n+1];

Edge e[]=new Edge[m+1];

Edge t=new Edge();

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

{

e[i]=new Edge();

System.out.println("enter 2 vertices and weights of edge:");

System.out.print("first vertex:");

e[i].u=s.nextInt();

System.out.print("second vertex");

e[i].v=s.nextInt();

System.out.print("weights:");

e[i].wt=s.nextInt();

}

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

{

for(j=i+1;j<=m;j++)

{

if(e[i].wt>e[j].wt)

{

t=e[i];

e[i]=e[j];

e[j]=t;

}

}

}

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

{

p[i]=-1;

}

i=1;

j=1;

mincost=0;

System.out.println();

while((i<=n-1) || (j<=m))

{

System.out.print("edge("+e[j].u+","+e[j].v+")"+"with weight"+e[j].wt+" ");

if(checkcycle(e[j]))

{

mincost=mincost+e[j].wt;

i++;

System.out.println("is Selected");

}

else

{

System.out.println("is discarded");

}

j++;

}

if(i!=n-1)

{

System.out.println("min spanning tree cannot be formed");

}

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

}

public static boolean checkcycle(Edge e)

{

int u=e.u,v=e.v;

while(p[u]>=0)

u=p[u];

while(p[v]>=0)

v=p[v];

if(u!=v)

{

p[u]=v;

return true;

}

return false;

}

static class Edge

{

int u,v,wt;

}

}