KRUSKAL ALGORITHM

```
import java.util.Scanner;
public class kruskal {
int parent[]=new int[10];
int find(int m)
{
int p=m;
while(parent[p]!=0)
p=parent[p];
return p;
}
void union(int i,int j)
{
if(i<j)
parent[i]=j;
else
parent[j]=i;
}
void krkl(int[][]a, int n)
{
int u=0,v=0,min,k=0,i,j,sum=0;
while(k<n-1)
{
min=99;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(a[i][j]<min&&i!=j)
{
min=a[i][j];
u=i;
v=j;
}
i=find(u);
j=find(v);
if(i!=j)
{
union(i,j);
System.out.println("("+u+","+v+")"+"="+a[u]
[v]);
sum=sum+a[u][v];
k++;
}
a[u][v]=a[v][u]=99;
}
System.out.println("The cost of minimum
spanning tree = "+sum);
}
public static void main(String[] args) {
int a[][]=new int[10][10];
int i,j;
System.out.println("Enter the number of
vertices of the graph");
Scanner sc=new Scanner(System.in);

int n;
n=sc.nextInt();
System.out.println("Enter the wieghted
matrix");
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
a[i][j]=sc.nextInt();
kruskal k=new kruskal();
k.krkl(a,n);
sc.close();
}
}
```