**ASSIGNMENT 5**

**AIM:**

You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures

**CODE:**

include<iostream>

using namespace std;

#define MAX 30

typedef struct edge

{

int u,v,w;

}edge;

typedef struct edgelist

{

edge data[MAX];

int count;

}edgelist;

edgelist elist;

int G[MAX][MAX],n;

edgelist spanlist;

void kruskal();

int find(int belongs[],int vertexno);

void union1(int belongs[],int c1,int c2);

void sort();

void print();

int main()

{

int i,j;

cout<<"\nEnter number of city's:";

cin>>n;

cout<<"\nEnter the adjacency matrix of city ID's:\n";

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

for(j=0;j<n;j++)

cin>>G[i][j];

kruskal();

print();

}

void kruskal()

{

int belongs[MAX],i,j,cno1,cno2;

elist.count=0;

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

for(j=0;j<i;j++)

{

if(G[i][j]!=0)

{

elist.data[elist.count].u=i;

elist.data[elist.count].v=j;

elist.data[elist.count].w=G[i][j];

elist.count++;

}

}

sort();

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

belongs[i]=i;

spanlist.count=0;

for(i=0;i<elist.count;i++)

{

cno1=find(belongs,elist.data[i].u);

cno2=find(belongs,elist.data[i].v);

if(cno1!=cno2)

{

spanlist.data[spanlist.count]=elist.data[i];

spanlist.count=spanlist.count+1;

union1(belongs,cno1,cno2);

}

}

}

int find(int belongs[],int vertexno)

{

return(belongs[vertexno]);

}

void union1(int belongs[],int c1,int c2)

{

int i;

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

if(belongs[i]==c2)

belongs[i]=c1;

}

void sort()

{

int i,j;

edge temp;

for(i=1;i<elist.count;i++)

for(j=0;j<elist.count-1;j++)

if(elist.data[j].w>elist.data[j+1].w)

{

temp=elist.data[j];

elist.data[j]=elist.data[j+1];

elist.data[j+1]=temp;

}

}

void print()

{

int i,cost=0;

for(i=0;i<spanlist.count;i++)

{

cout<<"\n"<<spanlist.data[i].u<<" "<<spanlist.data[i].v<<" "<<spanlist.data[i].w;

cost=cost+spanlist.data[i].w;

}

cout<<"\n\nMinimum cost of the telephone lines between the cities:"<<cost<<"\n";

}

OUTPUT:

