ANUSREE. P. R

AM. SC. P2CSC19011

**MINIMUM SPANNING TREE**

KRUSKAL’S ALGORITHM

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

int cost[10][10],i,j,k,n,m,c,visit,visited[10],l,v,count,count1,vst,p;

main()

{

int dup1,dup2;

cout<<"enter no of vertices";

cin >> n;

cout <<"enter no of edges";

cin >>m;

cout <<"EDGE Cost";

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

{

cin >>i >>j >>c;

cost[i][j]=c;

cost[j][i]=c;

}

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

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

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

cost[i][j]=31999;

visit=1;

while(visit<n)

{

v=31999;

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

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

if(cost[i][j]!=31999 && cost[i][j]<v && cost[i][j]!=-1 )

{

int count =0;

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

{

if(visited[p]==i || visited[p]==j)

count++;

}

if(count >= 2)

{

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

if(cost[i][p]!=31999 && p!=j)

dup1=p;

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

if(cost[j][p]!=31999 && p!=i)

dup2=p;

if(cost[dup1][dup2]==-1)

continue;

}

l=i;

k=j;

v=cost[i][j];

}

cout <<"edge from " <<l <<"-->"<<k;

cost[l][k]=-1;

cost[k][l]=-1;

visit++;

int count=0;

count1=0;

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

{

if(visited[i]==l)

count++;

if(visited[i]==k)

count1++;

}

if(count==0)

visited[++vst]=l;

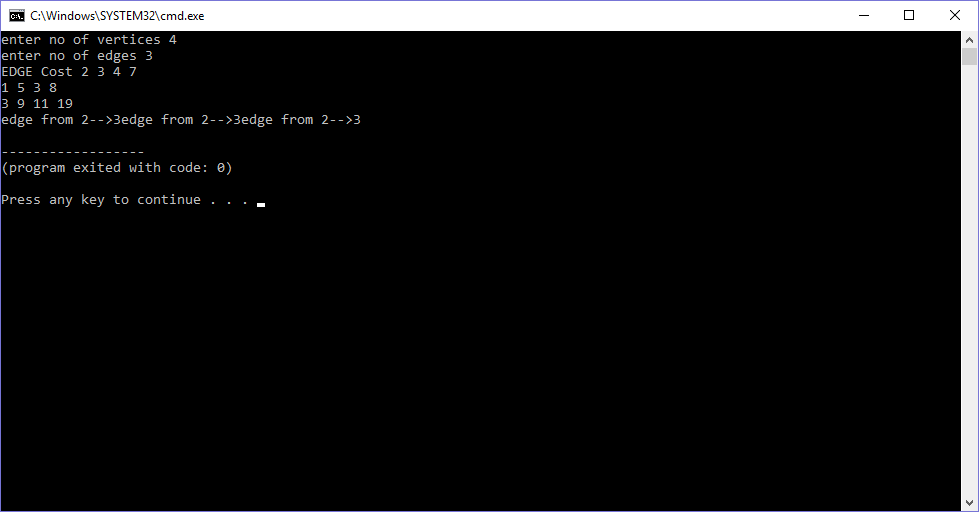
if(count1==0)

visited[++vst]=k;

}

}

OUTPUT



PRIMS ALGORITHM

#include <iostream>

#include <conio.h>

using namespace std;

struct node

{

int f, t, cost;

}p[6];

int c = 0, temp1 = 0, temp = 0;

void prims(int \*a, int b[][4], int i, int j)

{

a[i] = 1;

while (c < 3)

{

int min = 999;

for (int i = 0; i < 4; i++)

{

if (a[i] == 1)

{

for (int j = 0; j < 4; )

{

if (b[i][j] >= min || b[i][j] == 0)

{

j++;

}

else if (b[i][j] < min)

{

min = b[i][j];

temp = i;

temp1 = j;

}

}

}

}

a[temp1] = 1;

p[c].f = temp;

p[c].t = temp1;

p[c].cost = min;

c++;

b[temp][temp1] = b[temp1][temp]=1000;

}

for (int k = 0; k < 3; k++)

{

cout<<"source node:"<<p[k].f<<endl;

cout<<"destination node:"<<p[k].t<<endl;

cout<<"weight of node"<<p[k].cost<<endl;

}

}

int main()

{

int a[4];

for (int i = 0; i < 4; i++)

{

a[i] = 0;

}

int b[4][4];

for (int i = 0; i < 4; i++)

{

cout<<"enter values for "<<(i+1)<<" row"<<endl;

for (int j = 0; j < 4; j++)

{

cin>>b[i][j];

}

}

prims(a, b, 0, 0);

getch();

}

OUTPUT

