Program to find minimum cost spanning tree using Prim’s algorithm MCA584

Program code

#include<stdio.h>

#include<conio.h>

int n,cost[10][10];

void prim();

void main()

{

int i,j;

clrscr();

printf("\nEnter the number of vertices:");

scanf("%d",&n);

printf("\nEnter the costs of edges in matrix form:");

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

{

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

{

scanf("%d",&cost[i][j]);

}

}

printf("\n The matrix is:\n");

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

{

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

{

printf("%d ",cost[i][j]);

}

printf("\n\n");

}

prim();

getch();

}

void prim()

{

int i,j,k,l,x,nr[10],temp,mincost=0,tree[10][3];

temp=cost[0][0];

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

{

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

{

if(temp>cost[i][j])

{

temp=cost[i][j];

k=i;

l=j;

}

}

}

tree[0][0]=k;

tree[0][1]=l;

tree[0][2]=temp;

mincost=temp;

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

{

if(cost[i][k]<cost[i][l])

nr[i]=k;

else

nr[i]=l;

}

nr[k]=100;

nr[l]=100;

temp=99;

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

{

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

{

if(nr[j]!=100&&cost[j][nr[j]]<temp)

{

temp=cost[j][nr[j]];

x=j;

}

}

tree[i][0]=x;

tree[i][1]=nr[x];

tree[i][2]=cost[x][nr[x]];

mincost=mincost+cost[x][nr[x]];

nr[x]=100;

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

{

if(nr[j]!=100&&cost[j][nr[j]]>cost[j][x])

nr[j]=x;

}

temp=99;

}

printf("\nThe minimum cost spanning tree is:\n");

for(i=0;i<n-1;i++)

{

for(j=0;j<3;j++)

{

printf("%d ",tree[i][j]);

}

printf("\n\n");

}

printf("\nThe minimum cost:%d",mincost);

}

Output

Enter the number of vertices: 3

Enter the costs of edges in matrix form:

23 34 5

6 7 8

4 1 3

The matrix is:

23 34 5

6 7 8

4 1 3

The minimum cost spanning tree is:

2 1 1

0 2 5

The minimum cost: 6