MST

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

#define infinity 9999

#define MAX 20

int G[MAX][MAX],spanning[MAX][MAX],n;

int prims();

void main()

{

clrscr();

int i,j,total\_cost;

cout<<"Enter no. of vertices:";

cin>>n;

cout<<"\nEnter the adjacency matrix:\n";

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

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

{

cout<<"("<<i<<","<<j<<"): ";

cin>>G[i][j];

}

total\_cost=prims();

cout<<"\nspanning tree matrix:\n";

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

{

cout<<"\n";

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

cout<<spanning[i][j]<<" ";

}

cout<<"\n\nTotal cost of spanning tree=%d"<<total\_cost;

getch();

}

int prims()

{

int cost[MAX][MAX];

int u,v,min\_distance,distance[MAX],from[MAX];

int visited[MAX],no\_of\_edges,i,min\_cost,j;

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

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

{

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

cost[i][j]=infinity;

else

cost[i][j]=G[i][j];

spanning[i][j]=0;

}

distance[0]=0;

visited[0]=1;

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

{

distance[i]=cost[0][i];

from[i]=0;

visited[i]=0;

}

min\_cost=0;

no\_of\_edges=n-1;

while(no\_of\_edges>0)

{

min\_distance=infinity;

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

if(visited[i]==0&&distance[i]<min\_distance)

{

v=i;

min\_distance=distance[i];

}

u=from[v];

spanning[u][v]=distance[v];

spanning[v][u]=distance[v];

no\_of\_edges--;

visited[v]=1;

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

if(visited[i]==0&&cost[i][v]<distance[i])

{

distance[i]=cost[i][v];

from[i]=v;

}

min\_cost=min\_cost+cost[u][v];

}

return(min\_cost);

}

