Find Minimum Cost Spanning Tree of a given undirected graph using **Prim’s algorithm.**

Note: In the record book students should

- Handwrite the Algorithm,

- Handwrite the Program

- Pasting of the printout of the Output or handwrite the Output

**ALGORITHM:** prims(c[1…n,1…n])

//To compute the minimum spanning tree of a given weighted undirected graph using Prim’s

// algorithm

//Input: An nXn cost matrix c[1…n,1….n]

//Output: minimum cost of spanning tree of given undirected graph

ne🡨0

mincost🡨0

**for** i🡨1 to n **do**

elec[i]🡨1

**end for**

elec[1]🡨1

**while** ne!=n-1 **do**

min🡨9999

**for** i🡨1 to n **do**

**for** j🡨1 to n **do**

**if** elec[i]=1

**if** c[i,j]<min

min🡨c[i,j]

u🡨i

v🡨j

**end if**

**end if**

**end for**

**end for**

**if** elec[v]!=1

write u,v,min

elec[v]🡨1

ne🡨ne+1

mincost🡨mincost+min

**end if**

c[u,v]🡨9999

c[v,u]🡨9999

**end while**

write mincost

**return**

PROGRAM:

#include<stdio.h>

#include<conio.h>

#include<process.h>

void prims();

int c[10][10],n;

void main()

{

int i,j;

clrscr();

printf("\nenter the no. of vertices:\t");

scanf("%d",&n);

printf("\nenter the cost matrix:\n");

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

{

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

{

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

}

}

prims();

getch();

}

void prims()

{

int i,j,u,v,min;

int ne=0,mincost=0;

int elec[10];

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

{

elec[i]=0;

}

elec[1]=1;

while(ne!=n-1)

{

min=9999;

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

{

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

{

if(elec[i]==1)

{

if(c[i][j]<min)

{

min=c[i][j];

u=i;

v=j;

}

}

}

}

if(elec[v]!=1)

{

printf("\n%d----->%d=%d\n",u,v,min);

elec[v]=1;

ne=ne+1;

mincost=mincost+min;

}

c[u][v]=c[v][u]=9999;

}

printf("\nmincost=%d",mincost);

}

==========Output=============

Enter the no. of vertices: 6

Enter the cost matrix:

9999 3 9999 9999 6 5

3 9999 1 9999 9999 4

9999 1 9999 6 9999 4

9999 6 6 9999 8 5

6 9999 9999 8 9999 2

5 4 4 5 2 9999

2-----------> 3 = 1

5-----------> 6 = 2

1-----------> 2 = 3

2-----------> 6 = 4

4-----------> 6 = 5

Mincost = 15

Kruskal Algorithm

**AIM:** Find Minimum Cost Spanning Tree of a given undirected graph using **Kruskals algorithm**.

Note: In the record book students should

- Handwrite the Algorithm

- Handwrite the Program

- Pasting of the printout of the Output or handwrite the Output

**ALGORITHM:** kruskals(c[1…n,1…n])

//To compute the minimum spanning tree of a given weighted undirected graph using Kruskal’s

// algorithm

//Input: An nXn cost matrix c[1…n,1….n]

//Output: minimum cost of spanning tree of given undirected graph

ne🡨0

mincost🡨0

**for** i🡨1 to n **do**

parent[i]🡨0

**end for**

**while** ne!=n-1 **do**

min🡨9999

**for** i🡨1 to n **do**

**for** j🡨1 to n **do**

**if** c[i,j]<min

min🡨c[i,j]

u🡨i

a🡨i

v🡨j

b🡨j

**end if**

**end for**

**end for**

**while** parent[u]!=0 **do**

u🡨parent[u]

**end while**

**while** parent[v]!=0 **do**

v🡨parent[v]

**end while**

**if** u!= v

write a,b,min

parent[v]🡨u

ne🡨ne+1

mincost🡨mincost+min

**end if**

c[a,b]🡨9999

c[b,a]🡨9999

**end while**

write mincost

**return**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

void kruskals();

int c[10][10],n;

void main()

{

int i,j;

clrscr();

printf("\nenter the no. of vertices:\t");

scanf("%d",&n);

printf("\nenter the cost matrix:\n");

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

{

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

{

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

}

}

kruskals();

getch();

}

void kruskals()

{

int i,j,u,v,a,b,min;

int ne=0,mincost=0;

int parent[10];

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

{

parent[i]=0;

}

while(ne!=n-1)

{

min=9999;

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

{

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

{

if(c[i][j]<min)

{

min=c[i][j];

u=a=i;

v=b=j;

}

}

}

while(parent[u]!=0)

{

u=parent[u];

}

while(parent[v]!=0)

{

v=parent[v];

}

if(u!=v)

{

printf("\n%d----->%d=%d\n",a,b,min);

parent[v]=u;

ne=ne+1;

mincost=mincost+min;

}

c[a][b]=c[b][a]=9999;

}

printf("\nmincost=%d",mincost);

}

==========Output=============

Enter the no. of vertices: 6

Enter the cost matrix:

9999 3 9999 9999 6 5

3 9999 1 9999 9999 4

9999 1 9999 6 9999 4

9999 6 6 9999 8 5

6 9999 9999 8 9999 2

5 4 4 5 2 9999

2-----------> 3 = 1

5-----------> 6 = 2

1-----------> 2 = 3

2-----------> 6 = 4

4-----------> 6 = 5

Mincost = 15