**PROGRAM 14: Prim’s algorithm.**

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

**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;

int main()

{

int i,j;

printf("\n Enter the no. of vertices: ");

scanf("%d",&n);

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

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

{

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

{

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

}

}

prims();

}

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\t%c -----> %c = %d\n",u+65,v+65,min);

elec[v]=1;

ne=ne+1;

mincost=mincost+min;

}

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

}

printf("\n\n\t Minimum Cost is =%d\n",mincost);

}

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

Enter the no. of vertices:   6

Enter the cost matrix:

9999          3    9999    9999          6           5

      3    9999          1    6    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

B-----------> C = 1

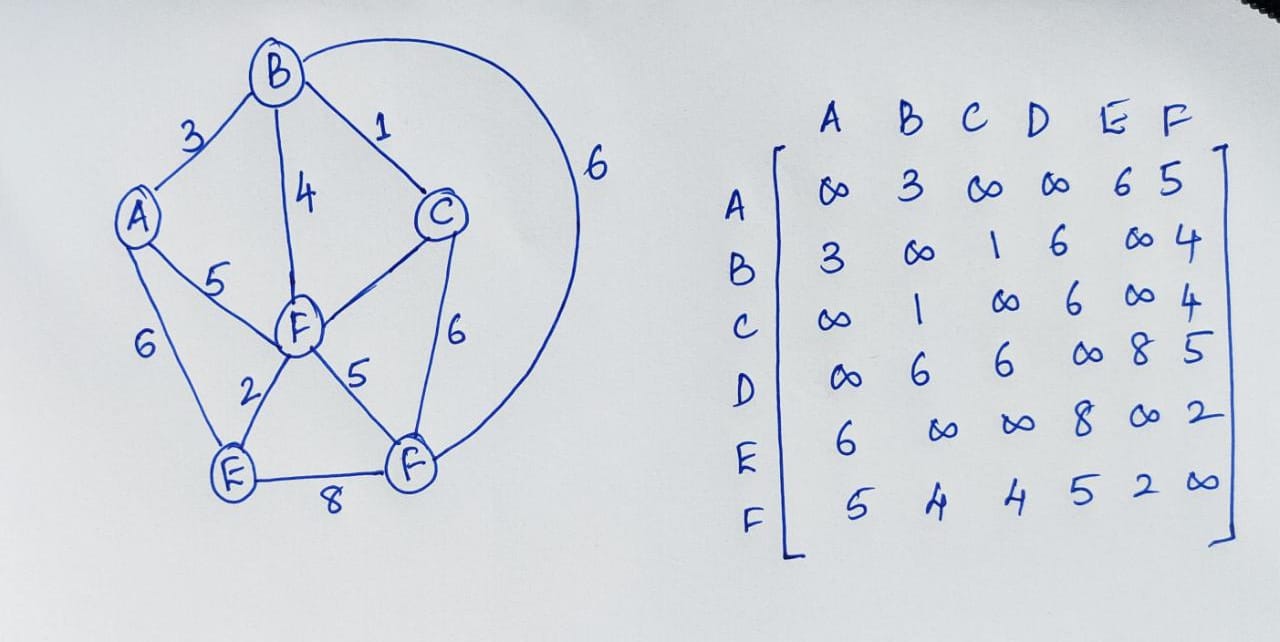
E-----------> F = 2

A-----------> B = 3

B-----------> F = 4

F-----------> D = 5

Minimum cost = 15





****

