**PROGRAM 15**

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

**Code:**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

int i,j,k,a,b,u,v,n,ne=1;

int min,mincost=0,cost[9][9],parent[9];

int find(int);

int uni(int,int);

void main()

{

double time;

clock\_t start,end;

printf("\nKruskal's algorithm\n");

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

scanf("%d",&n);

printf("\nEnter the cost adjacency matrix:\n");

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

{

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

{

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

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

cost[i][j]=999;

}

}

printf("The edges of Minimum Cost Spanning Tree are\n");

start=clock();

while(ne < n){

for(i=1,min=999;i<=n;i++)

{

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

{

if(cost[i][j] < min){

min=cost[i][j];

a=u=i;

b=v=j;}

}

}

u=find(u);

v=find(v);

if(uni(u,v))

{

printf("%d edge (%d,%d) =%d\n",ne++,a,b,min);

mincost +=min;

}

cost[a][b]=cost[b][a]=999;

}

end=clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nMinimum cost = %d\n",mincost);

printf("Time taken to find the minimum cost using the kruskal's algorithm is:%lf\n",time);

getch();

}

int find(int i)

{

while(parent[i])

i=parent[i];

return i;

}

int uni(int i,int j)

{

if(i!=j)

{

parent[j]=i;

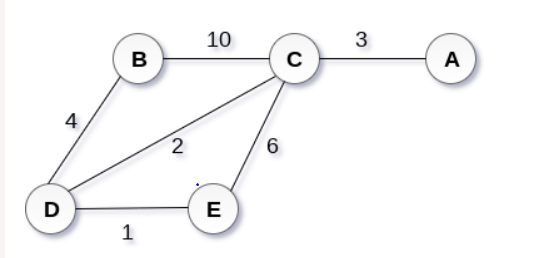
return 1;

}

return 0;

}

**Graph:**



**Adjacency matrix**

A B C D E

A 0 0 3 0 0

B 0 0 10 4 0

C 3 10 0 2 6

D 0 4 2 0 1

E 0 4 2 0 1

**Output:**

