

S.No: 16

Exp. Name: **Program to find minimum spanning tree of a given undirected graph using Kruskal's algorithm**

Date:

Aim:

Program to find minimum spanning tree of a given undirected graph using Kruskal's algorithm

Source Code:

kruskalsAlgorithm.c

```

#include<conio.h>
int parent[100];
int find(int i)
{
    while(parent[i]!=i)
        i=parent[i];
    return i;
}
void unio(int i,int j)
{
    int x,y;
    x=find(i);
    y=find(j);
    parent[x]=y;
}
void kruskal(int a[][100],int n)
{
    int k,co=0,min,r,b,l,res[100][2];
    for(k=0;k<n;k++)
        parent[k]=k;
    printf("The minimum spanning tree has the following edges:\n");
    while(co<n-1)
    {
        min=10000000;
        r=-1;
        b=-1;
        for(k=n-1;k>-1;k--)
        {
            for(l=n-1;l>-1;l--)
            {
                if(find(k)!=find(l) && a[k][l]<min && a[k][l]!=0)
                {
                    min=a[k][l];
                    r=k;
                    b=l;
                }
            }
        }
        unio(r,b);
        res[co][0]=r+1;
        res[co][1]=b+1;
        co++;
    }
    for(k=n-2;k>-1;k--)
        printf("%d-%d\n",res[k][0],res[k][1]);
}
void main()
{
    char c;
    int n,i,j,a[100][100],l[1000];
    printf("Input as adjacency matrix or adjacency list?(A/E)");
    scanf("%c",&c);

```

```
printf("no of nodes :");
scanf("%d",&n);
printf("Input as adjacency matrix:\n");
for(i=0;i<n;i++)
{
    printf("Row %d:",i+1);
    for(j=0;j<n;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
kruskal(a,n);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Input as adjacency matrix or adjacency list?(A/E) A
no of nodes : 6
Input as adjacency matrix: 0 3 1 6 0 0
Row 1: 0 3 1 6 0 0
Row 2: 3 0 5 0 3 0
Row 3: 1 5 0 5 6 4
Row 4: 6 0 5 0 0 2
Row 5: 0 3 6 0 0 6
Row 6: 0 0 4 2 6 0
The minimum spanning tree has the following edges:
6-3
2-1
5-2
6-4
3-1