

Find Minimum cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

```
#include <stdio.h>
void main (c)
{
    int a[20][20], b[20][20], c[20][20], d[20][20],
        nod=0, n, val=0, i, j, k, t, m=0, posx, posy, val;
    printf("\nEnter the value of n:");
    scanf ("%d", &n);
    printf ("\n Enter the adjacency matrix \n");
    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
        {
            scanf ("%d", &a[i][j]);
            b[i][j] = (i==j?0:a[i][j]);
            m=m+(b[i][j]?1:0);
            c[i][j]=0;
            d[i][j]=0;
        }
    }
    for (m=m/2; m!=0 && (nod!=(n-1)); m--)
    {
        val = 32767;
```

```
for (i=0; i<n; i++)  
{  
    for (j=0; j<n; j++)  
    {  
        if (b[i][j] != 0 && b[i][j] < val)  
        {  
            posx = i;  
            posy = j;  
            val = b[i][j];  
        }  
    }  
    b[posx][posy] = 0;  
    b[posy][posx] = 0;  
    if (c[posx][posy] == 0)  
    {  
        c[posx][posy] = 1;  
        c[posy][posx] = 1;  
        for (k=0; k<n; k++)  
        {  
            for (i=0; i<n; i++)  
            {  
                for (j=0; j<n; j++)  
                {  
                    c[i][j] = c[i][j] | (c[i][k] & c[k][j]);  
                }  
            }  
        }  
    }  
}
```

```
val 1 = val 1 + a[posx][posy];  
nod = nod + 1;  
d[posx][posy] = a[posx][posy];  
d[posy][posx] = a[posy][posy];  
}  
} → print("the node is", &nod);  
if (nod == n - 1)  
{  
    for (i = 0; i < n; i++)  
    {  
        printf("\n");  
        for (j = 0; j < n; j++)  
        {  
            printf("%d", d[i][j]);  
        }  
    }  
    printf("\n Spanning tree has a cost of %d", val1);  
}  
else  
{  
    printf("\n Spanning tree does not exist!");  
}  
}
```