MinSpanningTree_KruskalsAlgo.c

Saved

```c
#include<stdio.h>
void kruskals();
int cost[10][10],n,sum,min,i,j,count,k,u,v,parent[10],t[0][0];
void union_ij(int,int);
int find(int);
void main()
{
    printf("Enter the Number of Vertices:");
    scanf("%d", &n);
    printf("Enter the Cost of Adjacency Matrix:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d", &cost[i][j]);
        }
    }
    kruskals();
}
void kruskals()
{
    count=0;
    k=0;
    sum=0;
    for(i=0;i<n;i++)
        parent[i]=i;
    while(count!=n-1)
    {
        min=999;
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if(cost[i][j]<min && cost[i][j]!=0)
                {
                min=cost[i][j];
                u=i;
                v=j;
                }
            }
        }
        i=find(u);
        j=find(v);
        if(i!=j)
        {
            t[k][0]=u;
            t[k][1]=v;
            k++;
            count++;
            sum=sum+cost[u][v];
            union_ij(i,j);
        }
        cost[u][v]=cost[v][u]=999;
    }
    printf("Minimum Spanning Tree");
    for(i=0;i<=n-1;i++)
    {
        printf("\n%d□%d", t[i][0],t[i][1]);
        printf("\nTotal Cost= %d", sum);
    }
}
void union_ij(int i, int j)
{
    if(i<j)
        parent[j]=i;
    else
        parent[i]=j;
}
int find(int v)
{
    while(parent[v]!=v)
        v=parent[v];
    return v;
}
```