# 电网建设造价模拟系统

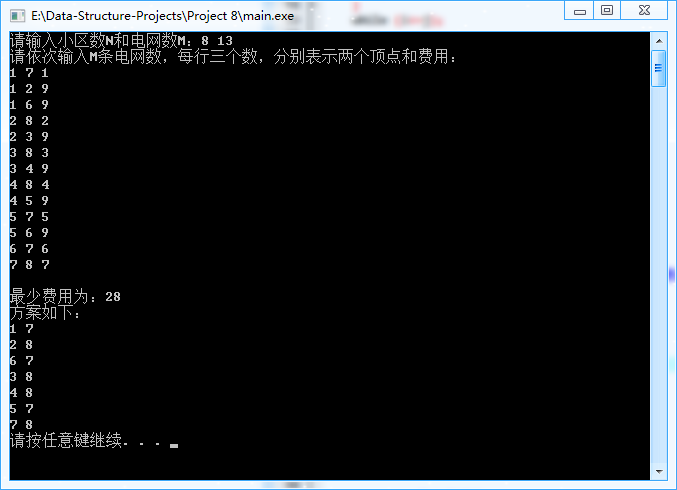
## 一.项目背景：

假设一个城市有n个小区，要实现n个小区之间的电网都能够相互接通，构造这个城市n个小区之间的电网，使总工程造价最低。请设计一个能够满足要求的造价方案。

## 二.项目功能

在每个小区之间都可以设置一条电网线路，都要付出相应的经济代价。n个小区之间最多可以有n（n-1）/2条线路，选择其中的n-1条使总的耗费最少。

项目示例：



## 三.程序描述

本题是一个裸的最小生成树

最小生成算法常见的有如下两种：

Prim算法 O(n^2) 🡺 堆优化后为O(nlogn)

Kruskal算法 O(mlogm)

我选择了Kruskal算法，因为该算法比较好写且时间复杂度很好

算法描述：

克鲁斯卡尔算法的时间复杂度为O（eloge）(e为网中边的数目)，因此它相对于普里姆算法而言，适合于求边稀疏的网的最小生成树。

克鲁斯卡尔算法从另一途径求网的最小生成树。假设连通网N=（V，{E}），则令最小生成树的初始状态为只有n个顶点而无边的非连通图T=（V，{∮}），图中每个顶点自成一个连通分量。在E中选择代价最小的边，若该边依附的顶点落在T中不同的连通分量上，则将此边加入到T中，否则舍去此边而选择下一条代价最小的边。依次类推，直至T中所有顶点都在同一连通分量上为止。

核心代码实现如下：

相关变量：

edge ele[1000];//边

int n,m,i,j,sum=0;

int fa[1000];//并查集记录父亲

bool flag[1000];//该边是否使用

首先是并查集：

//并查集找爸爸

int getFa(int x)

{

int fx;

fx=(x==fa[x])?x:getFa(fa[x]);

fa[x]=fx;//路径压缩

return fx;

}

//并查集合并

void uUnion(int x,int y)

{

int fx,fy;

fx=getFa(x);

fy=getFa(y);

fa[fy]=fx;

}

然后是快速排序qsort（省略，该算法很常见）

算法主体：

//克鲁斯卡尔算法

qsort(ele,0,m-1);

for (i=0;i<m;i++)

if (getFa(ele[i].x)!=getFa(ele[i].y))

{

sum+=ele[i].val;

uUnion(ele[i].x,ele[i].y);//合并

flag[i]=true;//使用了该边

}