**题目：图3、****Kruskal’s Algorithm for finding Minimum Spanning Tree**

**方法：**

用sort（）函数对所有的边按照权重从小到大排序，使用并查集原理，find（）函数查找到根节点，再把这一路的点进行路径压缩，判断（u,v）是否为同一棵树，若不是，则加入到最小生成树中。

**逻辑：**

①将边按照权重从小到大排列。

②枚举第一条边，用find（）函数查找u和v的根节点，判断加入最小生成树后是否成环。

③若u、v根节点相同，则成环，跳过这条边，否则确定这条边为最小生成树里的。

④继续枚举下一条边，直到所有的边都枚举完，留下n-1条边，n个点构成最小生成树。

**核心代码：**

void Kruskal()

{

sort(e + 1, e + m + 1, cmp);//根据权重从小到大排序

for (int i = 1; i <= m; i++)

{

int u = find(e[i].u);

int v = find(e[i].v);

if (u == v)

continue;//判断两个点是否在同一颗树,同一棵树则成环跳过

ans += e[i].w;

printf("%d->%d 权重:%d\n", u, v, e[i].w);

f[v] = u;//v点的根节点为u，(u,v)这条边加入

sum++;

if (sum == n - 1)break;//所有的点构成一棵树

}

}

**结果截图：**

