L题题解：最小生成树计数，缩点+矩阵树定理

首先，想要做这道题必须要知道的关于最小生成树的两大性质：

- 不同的最小生成树中,每种权值的边出现的个数是确定的；
- 不同的生成树中,某一种权值的边连接完成后,形成的联通块状态是一样的；

第一种性质简单说就是，假如某个最小生成树用了2个权值为1的边，3个权值为2的边，那么该图所有最小生成树用到的权值为1的边和权值为2的分均为2个和3个；

第二种性质简单说就是，相同权值的边无论怎样先后顺序如何，连接后的连通块完全一样。

比如说，假如三条权值为2的边（分别编号1，2，3），最小生成树中用到了两个权值为2的边，那么无论是用编号1，2还是1，3还是2，3，连接后形成的连通块都是一样的。

其次，还要利用矩阵树定理：

什么是矩阵树定理？~~具体证明这里就不解释了，想要解这个题目只要知道有这个结论就行了。~~

矩阵树定理可以被用来求生成树的数量。其可被描述为：

> G的所有不同的生成树的个数等于其Kirchhoff矩阵C[G]任何一个n-1阶主子式的行列式的绝对值。

G就是题目给我们的错综杂乱的图，n-1阶主子式大一上线代才学过 ~~（虽然我也已经忘得差不多了）~~。

而Kirchhoff矩阵，得名于证明矩阵树定理的那个人。关于它的构建方法如下：

- 定义G的度数矩阵D[G]为一个n*n的矩阵，并且满足：当i-j时,dij=0；当i=j时，dij等于vi的度数。
- 定义G的邻接矩阵A[G]为一个n*n的矩阵，并且满足：如果vi、vj之间有边直接相连，则aij=1，否则为0。
- 则Kirchhoff矩阵C[G]为C[G]=D[G]-A[G]。

关于行列式的计算，大多采用高斯消元法

```
//辗转相除法
int gauss(int num)
{
 int res=1;
 for(int i=1;i<num;++i)//从第一行计算到倒数第二行
 {
  for(int j=i+1;j<num;++j)//每次把第一行的对应元素变为0，然后操作行与第一行交换
  while(ma[j][i])
  {
   int t=ma[i][i]/ma[j][i];
   for(int k=i;k<num;++k)
   ma[i][k]-=t*ma[j][k];
   swap(ma[j],ma[i]);
   res=-res;//每次交换行列式的大小取负
  }
  res*=ma[i][i];//上三角矩阵的行列式的计算方法：对角元全部乘一遍
 }
 return res;
}
```

矩阵树定理是求生成树数量的，但是通过一些转化，便可以用来计算最小生成树的数量。这个转化就是缩点操作。

根据性质二，选定某个权重，将所有求最小生成树时用到的不是该选中的边都连上，此时整个图会形成数个连通块。

对这些连通块进行缩点操作，此时再运用矩阵树定理，就可以得到只考虑我们之前选中的那个权重可以得到的最小生成树的数量（边权之和第一次求得的最小生成树相同，所以这种方法求得的也是最小生成树）。

对于最小生成树中包含的每个权重都进行上次操作，因为每次上述操作都是独立的，所以可以运用乘法原理把以上结果乘起来，最终得到的就是最小生成树的数量。

下面对该题的算法进行个总结：

- 并查集+kruskal，找到一个最小生成树
- 同时记录各个权重，依次放到w_used[]中；记录使用的边，依此放到l_used[]中；
- 从第一种权重值循环到最后一种权重值，每次循环时，将除了被选中的权重的边连接起来（利用了w_used[]数组）
- 再利用并查集，缩点(每个点的代表储存在p_point[]中)，同时记录连通块的个数（缩成了几个点，单个的点也算）
- 对缩点后的图进行矩阵树定理
- 乘法原理

对要使用到的各个变量、数组进行解释

```
pre[]//用于并查集，将两个点归为一个集合
w_used[]//用于激励使用过的不同的权重（从小到大依此递增）
num_point//用于记录连通块的个数
p_point[]//用于缩点
ma[][]//最重要的那个矩阵
l_used[][]//用于记录生成的第一个最小生成树时使用的边
```

对使用到的函数进行解释

```
int find(int x)//并查集+路径压缩
bool cmp(Edge x,Edge y)//用于sort判断排序方法
void kruskal()//kruskal算法找到一组最小生成树的边
void con_line()//在kruskal算法中将出去选中的权值的边都连起来
void narrow_point()//缩点
void build_ma()//构建矩阵
int gauss(int num)//高斯消元，num是矩阵的维度
void solve()//整合上述函数到solve函数中
```

> **时间复杂度O（n^3):**
>
> 矩阵树定理.....高斯消元，学过线性代数算过高阶行列式的都懂......