

## I 题题解

### 【题目描述】

从前有个联通图，图上有好多边，边上有红绿两种颜色(好吧黑白其实也可以)，让你找个生成树里面两种颜色的边各占一半。

### 【算法分析】

拿并查集先把红边都连一块，然后在这个基础上加白边，然后去掉红边再在已有的白边上加白边加到一半为止(这时候加一条白边等于去掉一条红边)，然后再看情况把红边补上就行了。如果中间发现白边不够或者红边不够就无解。

### 【代码】

```
#include <iostream>
#include <string.h>
#include <cstdio>
#include <algorithm>
using namespace std;

const int N = 1200;
const int M = 120000;

struct edge{
    int u,v,k,flag;
}e[M];
int f1[N],f2[N];
int n,m;

int s1(int k){
    return f1[k]==k?k:f1[k]=s1(f1[k]);
}
int s2(int k){
    return f2[k]==k?k:f2[k]=s2(f2[k]);
}
void union1(int p,int q){
    f1[s1(p)]=s1(q);
}
void union2(int p,int q){
    f2[s2(p)]=s2(q);
}
void end(int k){
    if (k==1) puts("0");
    else puts("-1");
}

int main(){
    int p1,p2;
    char s[10];
```

```

while (scanf("%d%d",&n,&m)==2) {
    for (int i=1;i<=n;++i) f1[i]=f2[i]=i;
    for (int i=1;i<=m;++i) {
        scanf("%d%d%s",&e[i].u,&e[i].v,s);
        e[i].k=true;
        e[i].flag=false;
        if (s[0]=='S') {
            e[i].k=false;
            union1(e[i].u,e[i].v);
        }
    }
    if (n%2==0 || n==1) {
        end(n);
        continue;
    }
    int p1=0,p2=0;
    for (int i=1;i<=m;++i)
        if (e[i].k && s1(e[i].u)!=s1(e[i].v)) {
            union1(e[i].u,e[i].v);
            union2(e[i].u,e[i].v);
            e[i].flag=true;
            ++p1;
        }
    if (p1*2>n-1) {
        end(n);
        continue;
    }
    for (int i=1;i<=m && p1*2<n-1;++i)
        if (e[i].k && s2(e[i].u)!=s2(e[i].v)) {
            union2(e[i].u,e[i].v);
            e[i].flag=true;
            ++p1;
        }
    if (p1*2!=n-1) {
        end(n);
        continue;
    }
    for (int i=1;i<=m;++i)
        if (!e[i].k && s2(e[i].u)!=s2(e[i].v)) {
            union2(e[i].u,e[i].v);
            e[i].flag=true;
            ++p2;
        }
    if (p2*2!=n-1) {

```

```
        end(n);  
        continue;  
    }  
    printf("%d\n",n-1);  
    for (int i=1;i<=m;++i)  
        if (e[i].flag) printf("%d ",i);  
    puts("");  
}  
return 0;  
}
```