

# Codeforces Round #706 (Div. 1)

[比赛链接](#)

## C. Garden of the Sun

### 题意

给定一些黑格和一些白格，要求将一些白格转化为黑格，使得所有黑格连通但不出现环路。

题目保证以起始时以每个黑格为中心的  $3 \times 3$  范围内没有其他黑格。

### 题解

当  $n \equiv 1 \pmod{3}$  时，考虑将第  $1, 4, 7, \dots$  行染成黑色，然后对第  $3k+2, 3k+3, \dots$  行的每列，最多只有一个黑格。

如果第  $3k+2, 3k+3$  行间存在黑格，直接将两行的任意一个黑格所在列全染黑，否则将两行的第一列染黑。易知这样即可完成构造。

当  $n \not\equiv 1 \pmod{3}$  时，将  $1, 4, 7, \dots$  行换成第  $2, 5, 8, \dots$  行处理即可。

```
const int MAXN=505;
char buf[MAXN][MAXN];
int main()
{
    int T=read_int();
    while(T--){
        int n=read_int(), m=read_int();
        _for(i, 0, n) scanf("%s", buf[i]);
        int s1, s2;
        if(n%3==1){
            s1=0;
            s2=2;
        }
        else{
            s1=1;
            s2=3;
        }
        for(int i=s1; i<n; i+=3) _for(j, 0, m) buf[i][j]='X';
        for(int i=s2; i<n; i+=3){
            bool flag=false;
            _for(j, 0, m){
                if(buf[i-1][j]=='X' || buf[i][j]=='X'){
                    buf[i-1][j]=buf[i][j]='X';
                    flag=true;
                }
            }
        }
    }
}
```

```
        break;
    }
    if(!flag)
        buf[i-1][0]=buf[i][0]='X';
    _for(i,0,n)puts(buf[i]);
}
return 0;
}
```

## D. BFS Trees

### 题意

给定一个连通图，定义以点  $x$  为根的  $\text{BFS}$  树是生成树且树上所有点到  $x$  的距离等于连通图该点到  $x$  的距离。

定义  $f(x,y)$  表示既满足是以点  $x$  为根的  $\text{BFS}$  树同时也是以点  $y$  为根的  $\text{BFS}$  树的生成树的个数。

### 题解

首先给出结论：假定  $x, y$  之间有超过一条最短路径，则  $f(x,y)=0$

因为对于  $x, y$  之间最短路上的点  $u$  必有  $\text{dis}(x,u)+\text{dis}(u,y)=\text{dis}(x,y)$  易知树上满足该条件的点仅  $\text{dis}(x,y)+1$  个。

假如  $x, y$  之间有超过一条最短路径，则图中满足  $\text{dis}(x,u)+\text{dis}(u,y)=\text{dis}(x,y)$  的点必然超过  $\text{dis}(x,y)+1$  个，矛盾。

接下来仅考虑  $x, y$  之间仅有一条最短路的情况，首先易知生成树一定包含  $x, y$  之间的最短路。

接下来对除最短路外原图中的每条边  $u \rightarrow v$  假如保留该边，则必有  $\text{dis}(x,u)=\text{dis}(x,v) \pm 1, \text{dis}(y,u)=\text{dis}(y,v) \pm 1$

假如  $\text{dis}(x,u)+\text{dis}(u,y)=\text{dis}(x,v)+\text{dis}(v,y)$  则  $u, v$  必然在  $x \rightarrow y$  的路径上，矛盾。

于是为每个不在最短路上的结点指定一个父结点即可，答案即为所有点的所有可选父结点的个数的乘积。

设当前结点为  $u$  父结点为  $v$  则父结点应该满足  $\text{dis}(x,u)=\text{dis}(x,v)+1$  且  $\text{dis}(y,u)=\text{dis}(y,v)+1$

于是可以  $O(m)$  计算出每个  $f(x,y)$  总时间复杂度  $O(n^2m)$

```
const int MAXN=405,MAXM=605,Mod=998244353;
```

```

struct Edge{
    int to,next;
}edge[MAXM<<1];
int head[MAXN],edge_cnt,dis[MAXN][MAXN];
void Insert(int u,int v){
    edge[++edge_cnt]=Edge{v,head[u]};
    head[u]=edge_cnt;
}
int main()
{
    int n=read_int(),m=read_int();
    mem(dis,127/3);
    _rep(i,1,n)dis[i][i]=0;
    _for(i,0,m){
        int u=read_int(),v=read_int();
        dis[u][v]=dis[v][u]=1;
        Insert(u,v);Insert(v,u);
    }
    _rep(k,1,n)_rep(i,1,n)_rep(j,1,n)
    dis[i][j]=min(dis[i][j],dis[i][k]+dis[k][j]);
    _rep(x,1,n){
        _rep(y,1,n){
            int ans=1,cnt=0;
            _rep(u,1,n){
                if(dis[x][u]+dis[u][y]==dis[x][y])cnt++;
                else{
                    int cnt2=0;
                    for(int i=head[u];i;i=edge[i].next){
                        int v=edge[i].to;
                        if(dis[x][u]==dis[x][v]+1&&dis[y][u]==dis[y][v]+1)
                            cnt2++;
                    }
                    ans=1LL*ans*cnt2%Mod;
                }
            }
            space(cnt==dis[x][y]+1?ans:0);
        }
        puts(" ");
    }
    return 0;
}

```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:jxm2001:contest:cf\\_706\\_div.\\_1&rev=1616672006](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_706_div._1&rev=1616672006)

Last update: 2021/03/25 19:33