



[P4819][中山市选]杀人游戏

Algorithm	Tarjan 缩点 概率
Created	@Aug 9, 2020 2:45 PM
Difficulty	省选/NOI-
Related to 近期更新 (Property)	 [P4819][中山市选]杀人游戏
URL	https://www.luogu.com.cn/problem/P4819

画外音：好久没写题解了，AC了这题就来写题解。但是这道题真的让我好失望，不是我所想象的Tarjan建模题.....

题目链接：

[中山市选]杀人游戏

一位冷血的杀手潜入Na-wiat，并假装成平民。警察希望能在 N 个人里面，查出谁是杀手。警察能够对每一个人进行查证，假如查证的对象是平民，他会告诉警察，他认识的人，谁是杀手，谁是平民。假如查证的对象是杀手，杀手将会把警察干掉。现在警察掌握了每一个人认识谁。每一个人都有可能是杀手，可看作他

 <https://www.luogu.com.cn/problem/P4819>

题解：

这题不难。首先，假设我们已经查证了一个人并没有被杀，那么我们就可以知道他所有认识的人的信息。到这里，可以很自然地想到图论建模。每个人为一个节点，每个相识关系就是一条边。那么我们每过一个点就可以扩展到其相邻的所有点。而题目中要求让警察不被杀的可能性最大，因此就要使不确定查询（就是不确定查询对象是否是犯人的查询）的次数最小。那么我们可以求出图中的所有强连通分量，并缩点，然后找出所有入度为0的点，这就是最少不确定查询次数，并可以顺利求出概率了。

很快写完了代码，过了样例，兴冲冲地交上去，觉得可以AC，结果：21分？？？

我看了一下洛谷上的错误报告，发现我有输出为0.000000的情况，经分析，发现：

如果只有两个人，他们互相都不认识，查过一个人，发现他不是犯人，那么作为一个有脑子的警察，他可能再去查另外一个人去送死吗？

那么，也就是说，如果我们能找出这样一个“与世隔绝”的点（缩完点后的点），且其本身大小为1，那么，我们就可以不去查那个点。

问题来了，怎么定义“与世隔绝”呢？一个点是与世隔绝的充分必要条件就是：

1. 该点的大小为1
2. 该点的入度为0
3. 通过这个点所能到达的点的入度都不为1

大家可以自己思考一下其正确性，画个图看看就行了。

下面附上可爱的AC代码：

```
//File: P4819.cpp
//Author: yanyanlongxia
//Date: 2020/8/9
//[中山市选]杀人游戏
#include <bits/stdc++.h>

using namespace std;
int n,m,dfn[300005],low[300005],head[300005],nxt[300005],ver[300005],bel[300005],
sz[300005],sccn,tot,dft,din[300005];
stack<int>st;
bool in[300005];
double per;
int nver[300005],nnxt[300005],nhead[300005],ntot;//缩点后的新图
void add(int x,int y)
{
    ver[++tot]=y;
    nxt[tot]=head[x];
    head[x]=tot;
}
void new_add(int x,int y)
```

```

{
    nver[++ntot]=y;
    nnxt[ntot]=nhead[x];
    nhead[x]=ntot;
}
void tarjan(int x)
{
    dfn[x]=low[x]=++dft;
    st.push(x);
    in[x]=true;
    for(int i=head[x];i;i=nxt[i])
    {
        int y=ver[i];
        if(!dfn[y])
        {
            tarjan(y);
            low[x]=min(low[x],low[y]);
        }
        else
        {
            if(in[y])
                low[x]=min(low[x],dfn[y]);
        }
    }
    if(low[x]==dfn[x])
    {
        int y;
        sccn++;
        do {
            y=st.top();
            st.pop();in[y]=false;
            bel[y]=sccn;
            sz[sccn]++;//强连通分量的大小
        }while (x!=y);
    }
}
int main() {
    int x, y;
    scanf("%d%d", &n, &m);
    per = (double) 1 / n;//注意这里的1前面要加double
    for (int i = 1; i <= m; ++i) {
        scanf("%d%d", &x, &y);
        add(x, y);
    }
    for (int i = 1; i <= n; i++)
        if (!dfn[i])
            tarjan(i);
    for (int i = 1; i <= n; i++)
        for (int j = head[i]; j; j = nxt[j]) {
            if (bel[i] != bel[ver[j]]) {
                new_add(bel[i], bel[ver[j]]);//建立缩点图
                din[bel[ver[j]]]++;//统计入度
            }
        }
    int cnt = 0;
    bool p = 0;
    for (int i = 1; i <= sccn; ++i) {
        if (din[i] == 0) {

```

```

        cnt++; //统计入度为0的点
        if (sz[i] == 1) //大小为1
        {
            int ct1 = 0;
            for (int j = nhead[i]; j; j = nnxt[j])
            {
                int k = nver[j];
                if (din[k] == 1) //入度是否为1
                    ct1++;
            }
            if (ct1 == 0)
                p = 1;
        }
    }
    if (p)
        cnt--;
    double ans = (double)cnt * per;
    ans = (double)1 - ans;
    printf("%.6lf", ans);
    return 0;
}

```

PS: 我什么时候开启了狂刷水题模式.....