

[报告] G

[source]

<http://202.114.18.202:8080/judge/contest/view.action?cid=6190#problem/G>

[Description]

N 个牛，M 对关系，a b 表示 a 牛仰慕 b 牛，仰慕的关系具有传递性，问有多少个牛被其他所有牛仰慕。

[方法]

很明显在同一个强连通分量里的牛是互相仰慕的，于是先进行强连通分量缩点，然后在不同强连通分量里寻找出度为 0 的点（缩点之后的点），若出度为 0 的点只有 1 个，表明该强连通分量里的点的个数即为所求，如果出度为 0 的点超过 1 个，那么表明无解。

[Code]

```
#include<iostream>

#include<cstring>

#include<cstdio>

#include<vector>

#include<stack>

using namespace std;

#define N 10010

int DFN[N];

int LOW[N],col[N];
```

```

int instack[N];

int loc=0,scc=0;

stack<int> st;

vector<int> edge[N];

int deg[N];

void tarjan(int x)
{
    DFN[x]=LOW[x]=++loc;

    int u,v;

    st.push(x);

    instack[x]=1;

    for(int i=0;i<edge[x].size();i++){

        u=edge[x][i];

        if(!DFN[u]){

            tarjan(u);

            LOW[x]=min(LOW[x],LOW[u]);

        }

        else if(instack[u]) {

            LOW[x]=min(LOW[x],DFN[u]);

        }

    }

    if(DFN[x]==LOW[x]){

```

```

        scc++;

        do{

            v=st.top();

            col[v]=scc;

            instack[v]=0;

            st.pop();

        }while(v!=x);

    }

}

int main(void)

{

    int n,m,x,y;

    scanf("%d%d",&n,&m);

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

        scanf("%d%d",&x,&y);

        edge[x].push_back(y);

    }

    memset(DFN,0,sizeof(DFN));

    memset(deg,0,sizeof(deg));

    loc=0;scc=0;

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

    {

```

```

        if(!DFN[i])
            tarjan(i);
    }
    for(int i=1;i<=n;i++){
        for(int j=0;j<edge[i].size();j++) {
            int u=edge[i][j];
            if(col[i]!=col[u]){
                deg[col[i]]++;
            }
        }
    }

    int sum=0,mark;
    for(int i=1;i<=scc;i++){
        if(deg[i]==0) {
            sum++;
            mark=i;
        }
    }

    if(sum>1){
        printf("0\n");
    }

    else{

```

```
int ss=0;

for(int i=1;i<=n;i++){

    if(col[i]==mark)  ss++;

}

printf("%d\n",ss);

}

return 0;

}
```