**★★★★☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11504: Dominos**

**解題者：洪偉盛**

**解題日期：2022年5月12日**

**題意：**

給定一個骨牌遊戲，其中有一些骨牌間有關連，當其中一個倒了另一個一定也會倒，求需要最少用手推幾次才會全倒

**題意範例：**

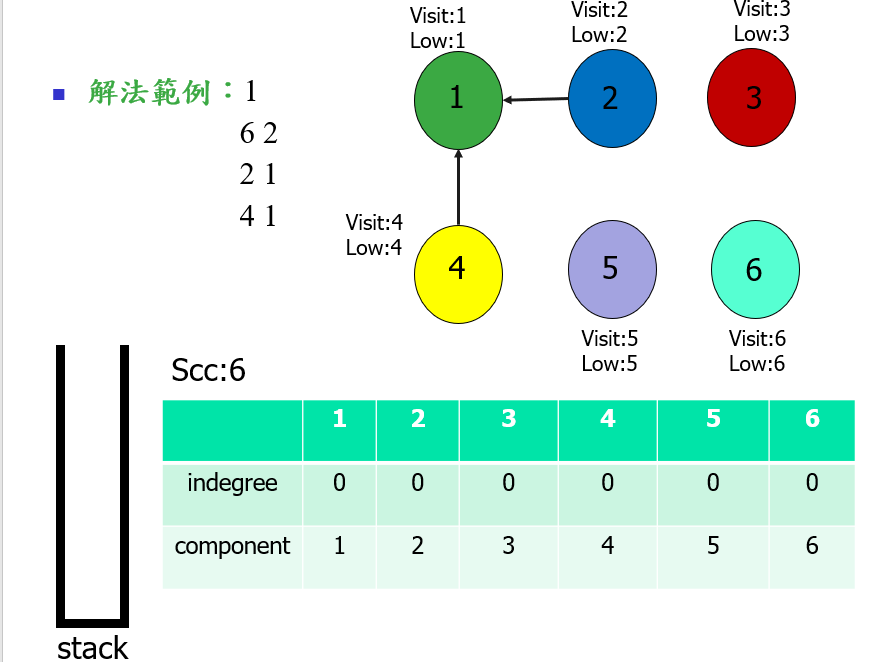
1🡺 幾筆測資(幾副骨牌)   
3 2🡺 3代表這副骨牌有三個骨牌，2代表有兩種關係  
1 2🡺 第一個倒了，第二個必倒

2 3 🡺 第二個倒了，第三個必倒

**解法：**

使用tarjan‘s 演算法來算SCC，再把每個SCC濃縮成一個點產生有向無環圖(DAG)，然後去找有多少入度為0的點就是代表手最少需要推多少下骨牌才會全倒

**解法範例：**



**討論：**

無

**程式：**

#include <iostream>

#include <stack>

#include <vector>

#include <cstdlib>

#include <cmath>

using namespace std;

#define N 100001

vector<int> edge[N]; //繪畫路徑

int visit[N];

int low[N];

int T,C;

stack<int> Stack;

bool instack[N];

int component[N];

int indegree[N];

int SCC;

int n , relation;

void init(void){ //初始化

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

edge[i].clear();

visit[i]=0;

instack[i]=false;

}

while(Stack.size()){

Stack.pop();

}

T=0;SCC=0;

}

void tarjan(int u){

Stack.push(u);

instack[u]=true;

visit[u] = low[u] = ++T;

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

int v = edge[u][i];

if(!visit[v]){

tarjan(v);

low[u] = min(low[u],low[v]);

}else if(instack[v]){

low[u] = min(low[u],visit[v]);

}

}

if(visit[u]==low[u]){

int t;

SCC++;

indegree[SCC]=0;

do{

t=Stack.top();

Stack.pop();

instack[t] = false;

component[t] = SCC;

}while(t!=u);

}

}

int main(){

int test;

cin>>test;

while(test>0){

cin>>n>>relation;

init(); //初始化

while(relation>0){

int x,y;

cin>>x>>y;

edge[x].push\_back(y); //繪畫路徑

relation--;

}

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

if(!visit[u]){

tarjan(u);

}

}

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

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

int v = edge[u][i];

if(component[u] != component[v]){ //是否為同一個SCC

indegree[component[v]]++;

}

}

}

int scc\_count = 0;

for(int c=1;c<=SCC;c++){

if(indegree[c]==0){

scc\_count++;

}

}

cout<<scc\_count<<endl;

test--;

}

return 0;

}