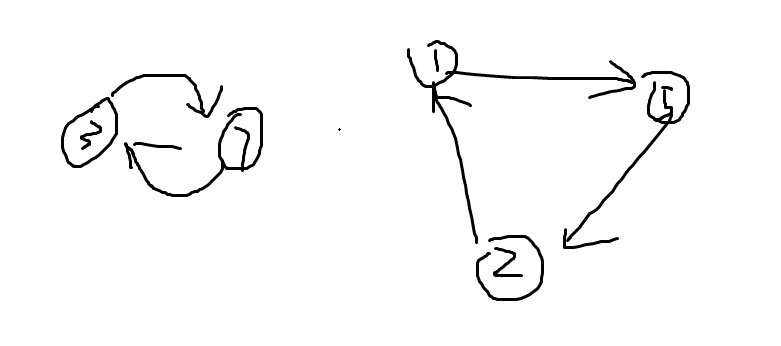


依然是属于有环，看成是有环的



和它的区别在于，它们都是有环，但图1只有一个顶点，而图二会有两个顶点

#include <cstdio>

#include <cstring>

#include<iostream>

#include<algorithm>

using namespace std;

typedef long long ll;

const int mod=1000000007;

int fa[1000010],num[1000010];

short b[1000010];

int n;

int Find(int x)

{

if(fa[x]==x)return x;

return fa[x]=Find(fa[x]);//是否需要带路径压缩还要考虑一下

}

void Merge(int x,int y)

{

x=Find(x);

y=Find(y);

if(x!=y)

{

fa[y]=x;

num[x]+=num[y];

b[x]|=b[y];//只要他要连上的集合是有环的，那么他就有环

}

else

b[x]=1;

}

int main()

{

//freopen("input.txt","r",stdin);

int x,y;

while(~scanf("%d",&n))

{

memset(fa,0,sizeof(fa));

memset(b,0,sizeof(b));

ll sum(1);

for(int i=1;i<=2\*n;i++)

{

fa[i]=i;

num[i]=1;//f[i]=点i的父节点,depth[i]=点i的深度,num[i]=i结点对应下面的元素个数

}

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

{

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

if(x==y)//自环

b[Find(x)]=2;//如果是自环 就不能移动了

else Merge(x,y);

}

for(int i=1;i<=2\*n;i++)

{

if(Find(i)==i)

{

if(b[i]==0)//无环，是棵树

sum=(sum\*num[i])%mod;

//无环就是整条链每次只可以换一个人的座位，有n个人在这个联通块

else

if(b[i]==1)//有环

sum=(sum\*2)%mod;

}

}

printf("%I64d\n",sum);

}

return 0;

}