

黑魔法师之门

一. 考察内容:

图论 连通性 树上问题

二. 题目分析:

[题目大意]

给出一个支持动态加边的图，求出每次操作后有多少闭合子图满足任意一个点的度数为正偶数。

[写题思路]

这道题有很多思考的方法，其实是一道结论题。

我们现在考察每个加边操作，首先将每个点放入图中，摆成树的模样（只是没有边）。对于一个加边操作，我们可以把它看成造这棵树，树有一个特征，任意两点之间不存在环，所以树上所有的边都不能构成答案，答案一定是非树边与树边一起贡献的。

考虑加入一条非树边（横叉边或返祖边），这时树上多了一个环，环上任意一点都是正偶数度，满足条件。考虑当前的答案数，我们可以从非树边中任意取出一些边加入闭合子图，都可以构建出一个合法的解，所以每增加一条非树边，答案都会变为原来的 2^{n+1} （不使用新边的答案数为 n ，这 n 种方案使用新边之后仍是答案，单独使用新边也有一种方案）。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 19
*文件类型: 源代码文件
*题目来源: COGS
*当前状态: 已通过
*备忘录: 树 计数问题
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 200001
#define MOD 1000000009
int n,m,f[MAXN];
namespace IO
{
#define SIZE 100000
int T=0,end=0;
char s[SIZE+1];
int in()
{
T=0;
return end=fread(s,1,SIZE,stdin);
}
int read(int &a)
{
int i=a=0;
```

```

        while(1)
        {
            if(T==end&&!in())return i;
            if(s[T]==' '||s[T]=='\n'||s[T]=='\r'){if(i)return i;}
            else a=a*10+s[T]-'0',++i;
            ++T;
        }
    }
}

/* Variable explain:

*/
using IO::read;
int mul(int a,int b)
{
    return (long long)a*b%MOD;
}
int add(int a,int b)
{
    return (a+b)%MOD;
}
void read()
{
    read(n);
    read(m);
    for(int i=1;i<=n;++i)f[i]=i;
    return;
}
int find(int x)
{
    return x==f[x]?x:f[x]=find(f[x]);
}
void merge(int a,int b)
{
    f[find(a)]=find(b);
}
int main()
{
    freopen("magiciiana.in","r",stdin);
    freopen("magiciiana.out","w",stdout);
    read();
    int ans=0;
    while(m--)
    {
        int ls1,ls2;
        read(ls1);
        read(ls2);
        if(find(ls1)==find(ls2))ans=add(mul(2,ans),1);
        else merge(ls1,ls2);
        printf("%d\n",ans);
    }
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)