# Polya

2888

#include<bits/stdc++.h>

using namespace std;

#define LL long long

int m,n;

const int mod=9973;

struct matrix{

int f[11][11];

};

int euler\_phi(int x){

int p=(int)sqrt(x+0.5);

int ans=x,i;

for(i=2;i<=p;i++)

if(x%i==0){

ans=ans/i\*(i-1);

while(x%i==0)x/=i;

}

if(x>1)ans=ans/x\*(x-1);

return ans%mod;// 注意mod，不然会爆int

}

matrix mul(matrix a,matrix b){

int i,j,k;

matrix c;

memset(c.f,0,sizeof(c.f));

for(k=1;k<=m;k++){

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

if(!a.f[i][k])continue;

for(j=1;j<=m;j++){

if(!b.f[k][j])continue;

c.f[i][j]=(c.f[i][j]+a.f[i][k]\*b.f[j][k])%mod;

}

}

}

return c;

}

matrix pow\_mod(matrix a,int b){

matrix s;

memset(s.f,0,sizeof(s.f));

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

s.f[i][i]=1;

while(b){

if(b&1)

s=mul(s,a);

a=mul(a,a);

b=b>>1;

}

return s;

}

int solve(matrix e,int x){

e=pow\_mod(e,x);

int i,ans=0;

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

ans=(ans+e.f[i][i])%mod;

return ans;

}

int pows(int a,int b){

int s=1;

while(b){

if(b&1)

s=(s\*a)%mod;

a=(a\*a)%mod;

b=b>>1;

}

return s;

}

int main(){

int T;

cin>>T;

while(T--){

int i,j,k,a,b,ans=0;

cin>>n>>m>>k;

matrix e;

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

for(j=1;j<=m;j++)

e.f[i][j]=1;

for(i=0;i<k;i++){

cin>>a>>b;

e.f[a][b]=e.f[b][a]=0;

}

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

if(n%i==0){

if(i\*i==n)

ans=(ans+euler\_phi(i)\*solve(e,i))%mod;

else

ans=(ans+euler\_phi(i)\*solve(e,n/i)+euler\_phi(n/i)\*solve(e,i))%mod;

}

}

cout<<ans\*pows(n%mod,mod-2)%mod<<endl;//pows里注意下n%mod

}

return 0;

}

/\*

这题是用欧拉函数，置换的Burnside引理和矩阵来解决的

欧拉函数euler\_phi(x)，求的是不超过x且与x互质的正整数个数

Burnside引理：对于一个置换f，若一个着色方案s经过置换后不变，称s为f的不动点。将f的不动点数记为C(f)，

则可以证明等价类数目为所有C(f)的平均值。

矩阵：

f[i][j]=1表示颜色i的后面可以接颜色j，而f[i][j]=0表示不行。得到矩阵A

A^k种∑f[i][i]表示长为k的符合要求的方案数，因为项链成环，第一个和第k+1个相同

sovle(k)表示长度为k的项链的方案数

现在讨论置换：这道题只用考虑旋转不用考虑翻转，所以有顺时针旋转1~n，共n种。

对于旋转i颗珠子，会出现p[i]=gcd(i,n)个循环，每个循环都是n/gcd(i,n);

找不动点，就是置换后各个位置的颜色都相同。所以第j个珠子和第j+p[i]个珠子的颜色相同。所以每p个珠子颜色重复一次，

因而只要求出p[i]个珠子长的项链有多少种方案，就是旋转i个珠子的不动点数

不动点总数ans=∑solve(p[i]),由于n可达1e9，所以不能直接求,由于p[i]=gcd(i,n)，所以p[i]只能是n的因子。

枚举因子，对于每个因子q，总共有euler\_phi(n/q)个不超过n的数与n的最大公约数是q；

因为只有当k<=n/q,且k与n/q互质的情况下，gcd(k\*q,n)=q;

最后ans=∑euler\_phi(n/q)\*slove(q)(q为n的因子)

答案就是(ans/n)%mod<-->ans\*n^(mod-2)%mod(mod为素数),这个可以用模乘法的逆来证明，可以直接当定理用了

\*/