polya定理

概念

设 $G=\{P_1,P_2,\ldots,P_g\}$ 是n个对象的一个置换群， $C(P_k)$ 是置换 P_k 的循环个数。用 m 种颜色对 n 个对象着色，着色方案数为 $\frac{1}{|G|}(m^{C(P_1)}+m^{C(P_2)}+\ldots+m^{C(P_g)})$

例题

洛谷p4980

首先由polya定理，首先存在 n 个置换群，分别对应着旋转0个，旋转1个….旋转 $n-1$ 个。然后我们不难得知，若旋转 k 个所对应的置换群为 P_k ，那么 $C(P_k)=gcd(k,n)$

所以答案就是 $\frac{1}{n}\sum_{k=1}^{n}n^{\gcd(k,n)}$

转变成枚举 $gcd(k,n)$ ，上式可化为 $\frac{1}{n}\sum_{d|n}n^d\phi(\frac{n}{d})$

时间复杂度 $O(Tn^{\frac{3}{4}})$

代码

```
#include <iostream>
#include <cmath>
#include <cstdio>
#define mod 1000000007
using namespace std;
inline long long fast_pow(long long a,long long b)
{
    long long an=1;
    long long now=a;
    for (;b;b>>=1,now=(now*now)%mod)
    if (b&1)
    an=(an*now)%mod;
    return an;
}
int phi(int x)
{
    int ans=x;
    for (int i=2;i<=sqrt(x);i++)
    if (x%i==0)
    {
        ans-=ans/i;
        while (x%i==0) x/=i;
```

```
    }
    if (x!=1) ans-=ans/x;
    return ans;
}
int main()
{
    int T;
    scanf("%d",&T);
    while (T--)
    {
        long long ans=0;
        int n;
        scanf("%d",&n);
        int cnt=sqrt(n);
        for (int i=1;i<=cnt;i++)
        if (n%i==0)
        {
            ans+=(fast_pow(n,n/i)*phi(i))%mod;
            ans%=mod;
            if (i*i!=n)
            {
                ans+=(fast_pow(n,i)*phi(n/i))%mod;
            ans%=mod;
            }
        }
        cout<<(1LL*ans*fast_pow(n,mod-2))%mod<<endl;
    }
    return 0;
 }
```