

## 问题描述 (洛谷4213)

给定 $N(N < 2^{31})$ ，求1到 $N$ 的欧拉函数和莫比乌斯函数之和

### 解决方法

#### 定义

对函数 $f(n), g(n)$ 定义 $(f * g)(n) = \sum_{d \mid n} f(d)g(\frac{n}{d})$ 为 $f$ 与 $g$ 的狄利克雷卷积

#### 引理1

$\sum_{i=1}^n (f * g)(i) = \sum_{i=1}^n \sum_{d \mid i} f(d)g(\frac{i}{d}) = \sum_{d=1}^n f(d) \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} g(i) = \sum_{d=1}^n f(d)S(\lfloor \frac{n}{d} \rfloor)$  其中 $S(n) = \sum_{i=1}^n g(i)$

#### 引理2

设 $f(n) = 1, g(n) = \phi(n)$  则有 $(f * g)(n) = n$

#### 引理3

设 $f(n) = 1, g(n) = \mu(n)$  则有 $(f * g)(n) = [n=1]$

### 具体解决

由引理1得 $f(1)S(n) = \sum_{i=1}^n (f * g)(i) - \sum_{d=2}^n f(d)S(\lfloor \frac{n}{d} \rfloor)$

分别将 $f(n) = 1, g(n) = \phi(n)$ 和 $f(n) = 1, g(n) = \mu(n)$ 带入，得

$S(n) = \frac{n^2 + n}{2} - \sum_{d=2}^n S(\lfloor \frac{n}{d} \rfloor)$  其中 $S(n) = \sum_{i=1}^n \phi(i)$

$T(n) = 1 - \sum_{d=2}^n T(\lfloor \frac{n}{d} \rfloor)$  其中 $T(n) = \sum_{i=1}^n \mu(i)$

先线性求出 $10^7$ 以内的 $S(n), T(n)$  对于大于 $10^7$ 的 $S(n), T(n)$  可通过数论分块递归计算出。

可以证明，杜教筛的时间复杂度为 $O(n^{\frac{3}{4}})$

```
#include<iostream>
#include<cstdio>
#include<map>
#define N 7000001
#define LL long long
```

```
using namespace std;

LL T,n,phi[N],mo[N],p[N>>3];
map<LL,LL>sphi,smo;
void initial()
{
    mo[1]=1; phi[1]=1;
    for(int i=2;i<N;i++)
    {
        if(!phi[i])
        {
            mo[i]=-1;
            phi[i]=i-1;
            p[++p[0]]=i;
        }
        for(int j=1;;j++)
        {
            LL t=i*p[j];
            if(t>=N) break;
            if(i%p[j]==0)
            {
                mo[t]=0;
                phi[t]=phi[i]*p[j];
                break;
            }
            mo[t]=-mo[i];
            phi[t]=phi[i]*(p[j]-1);
        }
    }
    for(int i=2;i<N;i++)
    {
        mo[i]+=mo[i-1];
        phi[i]+=phi[i-1];
    }
}
LL getphi(LL x)
{
    if(x<N) return phi[x];
    if(sphi[x]) return sphi[x];
    LL res=x*(x+1)>>1;
    for(LL l=2,r;l<=x;l=r+1)
    {
        r=x/(x/l);
        res-=(r-l+1)*getphi(x/l);
    }
    sphi[x]=res;
    return res;
}
LL getmo(LL x)
{

```

```
if(x<N) return mo[x];
if(smo[x]) return smo[x];
LL res=1;
for(LL l=2,r;l<=x;l=r+1)
{
    r=x/(x/l);
    res-=(r-l+1)*getmo(x/l);
}
smo[x]=res;
return res;
}
int main()
{
    initial();
    scanf("%lld",&T);
    while(T--)
    {
        scanf("%lld",&n);
        printf("%lld ",getphi(n));
        printf("%lld\n",getmo(n));
    }
    return 0;
}
```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:hotpot:%E6%9D%9C%E6%95%99%E7%AD%9B&rev=1590142020>

Last update: 2020/05/22 18:07