

数论 2

前置公式

$$\begin{aligned} \left(\sum_{d|n} f(d) \right) \left(\sum_{d|m} g(d) \right) &= \sum_{d_1|n} \sum_{d_2|m} f(d_1)g(d_2) \end{aligned}$$

$$\sum_{m=1}^n \sum_{d|(n,m)} f(d) = \sum_{d|n} \frac{f(d)n}{d}$$

$$\sum_{x|n} \left(\sum_{y|nx} g(y) \right) = \sum_{y|n} \left(\sum_{x|ny} f(x) \right)$$

可积函数

一般可积函数

定义

数论函数 θ 被定义为可积函数，若

$$(a) \exists n \text{ s.t. } \theta(n) \neq 0$$

$$(b) \forall n \forall m \theta(nm) = \theta(n)\theta(m)$$

性质

假定 θ 为可积函数，则

$$\begin{aligned} (1) \quad \theta(1) &= 1 \\ (2) \quad \theta(n) &= \theta(p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}) \\ (3) \quad \theta_1 \theta_2 \text{ 可积} &\Rightarrow \theta_1 \theta_2 \text{ 可积} \end{aligned}$$

定理

若 θ 为可积函数，则

$$\begin{aligned} (1) \quad \psi(n) &= \sum_{d|n} \theta(d) \\ (2) \quad \psi(n) &= \prod_{i=1}^k (1 + \theta(p_i) + \theta(p_i^2) + \dots + \theta(p_i^{\alpha_i})) \end{aligned}$$

证明

设 $(n, m) = 1$ 则对 $d \mid nm$ 一定存在唯一 $d_1 \mid n, d_2 \mid m, d = d_1 d_2$

同时，对每对 $d_1 \mid n, d_2 \mid m$ 一定有 $d = d_1 d_2 \mid nm$

所以根据 \$(1)\$ 式

$$\begin{aligned} \psi(nm) &= \sum_{d \mid nm} \theta(d) = \sum_{d_1 \mid n} \sum_{d_2 \mid m} \theta(d_1 d_2) \\ \theta(d_1) \theta(d_2) &= \left(\sum_{n \mid d_1} \theta(n) \right) \left(\sum_{m \mid d_2} \theta(m) \right) = \psi(n) \psi(m) \end{aligned}$$

$$\begin{aligned} \psi(n) &= \prod_{i=1}^k \psi(p_k^{\alpha_k}) = \prod_{i=1}^k \left(1 + \theta(p_k) \right) \\ &= \left(1 + \theta(p_1) \right) \left(1 + \theta(p_2) \right) \dots \left(1 + \theta(p_k^{\alpha_k}) \right) \tag{4} \end{aligned}$$

莫比乌斯函数

定义

$$\mu(n) = \begin{cases} 1 & n=1 \\ (-1)^k & n=p_1 p_2 \dots p_k \\ 0 & \text{otherwise} \end{cases}$$

性质

若 θ 为可积函数，根据 \$(4)\$ 式

$$\begin{aligned} \sum_{d \mid n} \mu(d) \theta(d) &= \prod_{i=1}^k \left(1 + \mu(p_k) \theta(p_k) \right) \\ &= \prod_{i=1}^k \left(1 + \mu(p_k) \left(1 - \theta(p_k) \right) \right) = \prod_{i=1}^k \left(1 - \mu(p_k) \theta(p_k) \right) \end{aligned}$$

特别地

$$\theta(n) \equiv 1 \iff \sum_{d \mid n} \mu(d) = \begin{cases} 1 & n=1 \\ 0 & \text{otherwise} \end{cases}$$

$$\theta(n) = \frac{1}{n} \sum_{d \mid n} \frac{\mu(d)}{d} = \frac{(1-p_1)(1-p_2)\dots(1-p_k)}{n}$$

欧拉函数

定义

$$\varphi(n) = \sum_{m=1}^n [(n, m) == 1]$$

性质

根据 \$(2)\$ 式、\$(6)\$ 式和 \$(7)\$ 式

$$\begin{aligned} \varphi(n) &= \sum_{m=1}^n [(\text{gcd}(n,m)=1)] = \sum_{m=1}^n \sum_{d|\text{gcd}(n,m)} \mu(d) \\ d &= n \sum_{d|n} \frac{\mu(d)}{d} = n(1-p_1)(1-p_2)\dots(1-p_k) \end{aligned}$$

其他常见可积函数

$$\tau(n) = \sum_{d|n} 1 = \prod_{i=1}^k (\alpha_k + 1)$$

$$\sigma(n) = \sum_{d|n} d = \prod_{i=1}^k (1 + p_k + p_k^2 + \dots + p_k^{\alpha_k})$$

完全积性函数

$f(x)$ 被定义为完全积性函数若 $\forall n \forall m (f(nm) = f(n)f(m))$

$$e(n) = [n=1]$$

$$I(n) = 1$$

$$id(n) = n$$

线性筛法

先给出线性筛素数的代码

```
const int MAXP=1e6;
bool vis[MAXP];
int prime[MAXP], cnt;
void Prime(){
    vis[1]=true;
    for(i, 2, MAXP){
        if(!vis[i]) prime[cnt++]=i;
        for(int j=0; j<cnt&&i*prime[j]<MAXP; j++){
            vis[i*prime[j]]=true;
            if(i%prime[j]==0) break;
        }
    }
}
```

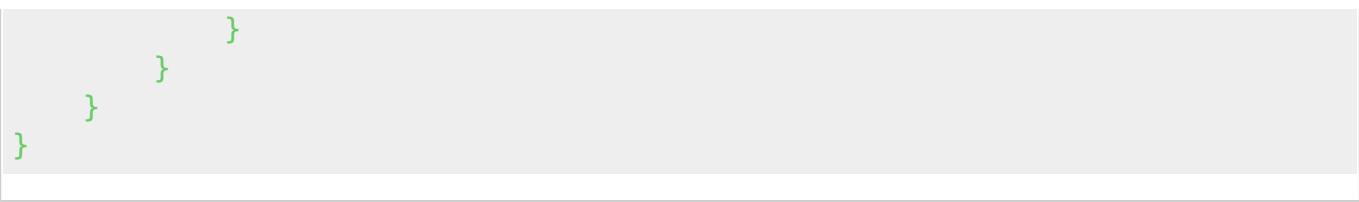
基于线性筛素数和可积函数性质，如果能在 $O(1)$ 时间求出 $f(p^k)$ 就可以线性筛 f 函数，例如

```
const int MAXP=1e6;
bool vis[MAXP];
int prime[MAXP], mu[MAXP], cnt;
void Mu(){
```

```
vis[1]=true, mu[1]=1;
_for(i,2,MAXP){
    if(!vis[i])mu[i]=-1, prime[cnt++]=i;
    for(int j=0;j<cnt&&i*prime[j]<MAXP;j++){
        vis[i*prime[j]]=true;
        if(i%prime[j])
            mu[i*prime[j]]=-mu[i];
        else{
            mu[i*prime[j]]=0;
            break;
        }
    }
}
}
```

```
const int MAXP=1e6;
bool vis[MAXP];
int prime[MAXP], phi[MAXP], cnt;
void Phi(){
    vis[1]=true, phi[1]=1;
    _for(i,2,MAXP){
        if(!vis[i])phi[i]=i-1, prime[cnt++]=i;
        for(int j=0;j<cnt&&prime[j]*i<MAXP;j++){
            vis[i*prime[j]]=true;
            if(i%prime[j])
                phi[i*prime[j]]=phi[i]*(prime[j]-1);
            else{
                phi[i*prime[j]]=phi[i]*prime[j];
                break;
            }
        }
    }
}
```

```
const int MAXP=1e6;
bool vis[MAXP];
int prime[MAXP], tau[MAXP], mpow[MAXP], cnt;
void Tau(){
    vis[1]=true, tau[1]=1;
    _for(i,2,MAXP){
        if(!vis[i])tau[i]=2, mpow[i]=1, prime[cnt++]=i;
        for(int j=0;j<cnt&&prime[j]*i<MAXP;j++){
            vis[i*prime[j]]=true;
            if(i%prime[j])
                tau[i*prime[j]]=tau[i]<<1, mpow[i*prime[j]]=1;
            else{
                tau[i*prime[j]]=tau[i]/(mpow[i]+1)*(mpow[i]+2), mpow[i*prime[j]]=mpow[i]+1;
                break;
            }
        }
    }
}
```



简单地说，

$$\begin{aligned} f(p) = \begin{cases} f(i)f(p) & p \text{ 不是 } i \text{ 的最小素因子} \\ f(i)\frac{f(p^{k+1})}{f(p^k)} & p \text{ 是 } i \text{ 的最小素因子} \end{cases} \end{aligned}$$

如有必要，额外存储每个数的最小素因子的幂次即可。

狄利克雷卷积

定义

两个数论函数 f 和 g 的迪利克雷卷积运算为 $\text{last } g(n) = \sum_{d \mid n} f(d) \text{last } g\left(\frac{n}{d}\right)$

性质

- 1、交换律 $\text{last } g = g \text{last } f$
- 2、结合律 $\text{last } (g \text{last } h) = (\text{last } g) \text{last } h$
- 3、分配律 $(f+g) \text{last } h = f \text{last } h + g \text{last } h$
- 4、单位元 $e \text{last } f = f$
- 5、逆元：对每个 $f(1) \neq 0$ 的数论函数 f 一定存在某个数论函数 g 使得 $\text{last } g = e$
- 6、两个可积函数的迪利克雷卷积仍为可积函数，可积函数的逆元也为可积函数。

性质 5 证明

事实上，构造 $g(n) = \frac{1}{f(1)} \left(e(n) - \sum_{d \mid n, d \neq 1} f(d) \text{last } g\left(\frac{n}{d}\right) \right)$

知 $g(n)$ 可由 $g(1 \sim n-1)$ 递推得到，且 $\sum_{d \mid n} f(d) \text{last } g\left(\frac{n}{d}\right) = f(1) \text{last } g(n) + \sum_{d \mid n, d \neq 1} f(d) \text{last } g\left(\frac{n}{d}\right) = e(n)$

莫比乌斯反演定理

设 $F(x) = f(x)$ 为数论函数 $F(n) = \sum_{d \mid n} f(d)$ iff $f(n) = \sum_{d \mid n} \mu(d)F\left(\frac{n}{d}\right)$

该定理的另一种形式为 $\begin{aligned} F(n) = \sum_{d|n} f(d) \iff f(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) F(d) \end{aligned}$

事实上，该定理等价于若 $F = \ast \mu$ 则 $f = F \ast \mu$ 。简单地说，莫比乌斯反演定理即证明 $e = \ast \mu$

证明

根据 \$(3)\$ 式和 \$(6)\$ 式

$$\begin{aligned} \sum_{d|n} \mu(d) F\left(\frac{n}{d}\right) &= \sum_{d|n} \left(\sum_{d_1|d} \mu(d_1) \sum_{d_2|\frac{n}{d_1}} f(d_2) \right) \\ &= \sum_{d_1|n} \left(\sum_{d_2|\frac{n}{d_1}} f(d_2) \right) \mu\left(\frac{n}{d_1}\right) = f(n) \end{aligned}$$

一些常见的狄利克雷卷积

$$1 \ast \mu = e$$

$$2 \ast \mu \ast id = \varphi$$

$$3 \ast id = \sigma$$

$$4 \ast id = \tau$$

$$5 \ast \varphi = id$$

$$6 \ast \tau \ast \varphi = \sigma$$

证明 1

莫比乌斯反演定理已证明。

证明 2

欧拉函数性质推导过程已证明。

证明 3

根据定义。

证明 4

根据定义。

证明 5

根据前面结论 $\ast \varphi = \ast (\mu \ast id) = (\ast \mu) \ast id = e \ast id = id$

证明 6

根据前面结论 $\ast \tau \varphi = (\ast I) \ast (\mu \ast id) = (\ast \mu) \ast (\ast id) = e \ast \sigma = \sigma$

算法练习

习题1

[洛谷p3455](#)

题意

给定 a, b, d 求满足 $x \leq a, y \leq b, (x, y) = d$ 的二元组。

题解 1

在约束条件 $i \leq a, j \leq b$ 下，设 $f(n)$ 为满足 $(i, j) = n$ 的二元组个数 $F(n)$ 为满足 $n \mid (i, j)$ 的二元组个数。

$$\begin{aligned} f(n) &= \sum_{i=1}^a \sum_{j=1}^b [i \mid j] = \lfloor \frac{a}{n} \rfloor \end{aligned}$$

$$\begin{aligned} F(n) &= \sum_{i=1}^a \sum_{j=1}^b [n \mid (i, j)] = \lfloor \frac{a}{\lfloor \frac{n}{i} \rfloor} \rfloor \end{aligned}$$

根据莫比乌斯反演定理，有

$$\begin{aligned} Ans &= f(d) = \sum_{d \mid k} \mu(k) \cdot f(\frac{k}{d}) \\ F(k) &= \sum_{d \mid k} \mu(k) \cdot f(\frac{k}{d}) \end{aligned}$$

设 $k=t \ast d$ 改为枚举 t 有

$$\begin{aligned} \sum_{d \mid k} \mu(k) \cdot f(\frac{k}{d}) &= \sum_{t=1}^{d-1} \mu(t) \cdot f(\frac{dk}{t}) \\ &= \sum_{t=1}^{d-1} \mu(t) \cdot \lfloor \frac{a}{\lfloor \frac{dk}{t} \rfloor} \rfloor \end{aligned}$$

剩下部分整数分块即可，时间复杂度 $O(\sqrt{d})$

```
#include <cstdio>
#include <iostream>
#include <vector>
#include <algorithm>
```

```
#include <cstring>
#include <cctype>
#include <queue>
#include <cmath>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset((a),(b),sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x) return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAXP=5e4+5;
bool vis[MAXP];
int prime[MAXP],mu[MAXP],cnt;
void Mu(){
    vis[1]=true,mu[1]=1;
    _for(i,2,MAXP){
        if(!vis[i])mu[i]=-1,prime[cnt++]=i;
        for(int j=0;j<cnt&&i*prime[j]<MAXP;j++){
            vis[i*prime[j]]=true;
            if(i%prime[j])
                mu[i*prime[j]]=-mu[i];
            else{
                mu[i*prime[j]]=0;
                break;
            }
        }
    }
}
int Ceil(int x,int y){return x%y?x/y+1:x/y;}
LL cal(int i,int n,int m,int k){
```

```
LL ans=0;
int lef=1,rig=0,tlef=Ceil(lef,k),trig;
while(tlef*k<=i){
    rig=min(i,min(n/(n/lef),m/(m/lef)));
    trig=rig/k;
    if(trig>=tlef)
        ans+=1LL*(mu[trig]-mu[tlef-1])*(n/lef)*(m/lef);
    lef=rig+1;
    tlef=Ceil(lef,k);
}
return ans;
}
int main()
{
    Mu();
    _for(i,2,MAXP)
    mu[i]+=mu[i-1];
    int t=read_int(),a,b,d;
    while(t--){
        a=read_int(),b=read_int(),d=read_int();
        enter(cal(min(a,b),a,b,d));
    }
    return 0;
}
```

题解 2

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E6%95%B0%E8%AE%BA_2&rev=1593828451

Last update: 2020/07/04 10:07

