

# 莫比乌斯反演

## 积性函数

来我们写一下这个部分……（虚晃一枪）啊好，没写呢

数论函数是一类定义域在正整数上的函数。若对数论函数  $f$  且  $\forall a, b$  使得  $(a, b) = 1$  都满足

$$f(ab) = f(a) \cdot f(b)$$

则称  $f$  是积性函数。如果条件弱一些，不需要  $(a, b) = 1$  也有上式成立，则称  $f$  是完全积性函数。只要  $f$  是一个积性函数，同时能够快速地求出  $f$  在质数  $p$  幂次上的取值  $f(p^a)$ ，那么就可以用线筛求  $f$

## 性质

若  $f, g$  都是积性函数，则以下函数也是积性函数：

$$h(n) = f(n^p) \quad h(n) = f^p(n) \quad h(n) = f(n)g(n) \quad h(n) = \sum_{d|n} f(d)g\left(\frac{n}{d}\right)$$

## 常见积性函数

定义艾佛森括号

$$[P] = \begin{cases} 1, & P \text{ true} \\ 0, & P \text{ false} \end{cases}$$

- 单位函数  $\varepsilon(n) = [n = 1]$
- 恒等函数  $\mathrm{id}(n) = n$  下文中一般会直接用  $n$  代替。
- 常数函数  $1(n) = 1$  下文中一般会直接用  $1$  代替。
- 欧拉函数  $\varphi(n) = \sum_{1 \leq i \leq n} [\gcd(i, n) = 1]$
- 约数函数  $d(n) = \sum_{d|n} 1$
- 约数和函数  $\sigma(n) = \sum_{d|n} d$

## 狄利克雷卷积

来我们写一下这个部分……（虚晃一枪）啊好，没写呢

对数论函数  $f, g$  定义它们的狄利克雷卷积

$$(f * g)(n) = \sum_{d|n} f(d)g\left(\frac{n}{d}\right)$$

## 性质

狄利克雷卷积满足：

- 交换律
- 结合律
- 分配律
- 单位元

同时，两个积性函数的狄利克雷卷积还是积性函数。

## 常用狄利克雷卷积关系

一个常用的卷积式子  $n = \varphi \times 1$  简单证明如下：

首先枚举约数，每个约数求出小于他且与他互质的个数，即求这个约数为分母的真分数个数，它们的和必为  $n$  例如  $n = 12$  时的真分数有

$$\frac{1}{12}, \frac{1}{12}, \frac{1}{12}$$

可以化简为

- $\frac{1}{12}, \frac{5}{12}, \frac{7}{12}, \frac{11}{12} \Rightarrow \varphi(12) = 4$
- $\frac{1}{6}, \frac{5}{6} \Rightarrow \varphi(6) = 2$
- $\frac{1}{4}, \frac{3}{4} \Rightarrow \varphi(4) = 2$
- $\frac{1}{3}, \frac{2}{3} \Rightarrow \varphi(3) = 2$
- $\frac{1}{2} \Rightarrow \varphi(2) = 1$
- $\frac{1}{1} \Rightarrow \varphi(1) = 1$

## 整数格上的莫比乌斯反演

### 莫比乌斯函数

莫比乌斯反演是偏序集上的一个反演，不过在此处我们只讨论整数格上的莫比乌斯反演。

定义莫比乌斯函数  $\mu(n)$

$$\mu(n) = \begin{cases} 1 & \forall (-1)^m, \& n = \prod_{i=1}^m p_i^{k_i}, \& \prod_{i=1}^m k_i = 1, \& p_i \text{ is a prime} \\ 0 & \& \& \& \text{otherwise} \end{cases}$$

$\mu(n)$  有两个性质：

- $\mu$  是积性函数
- $\sum_{d \mid n} \mu(d) = [n = 1]$

第一条性质说明  $\mu(n)$  可以线性筛；第二条性质提供了我们一个当且仅当  $n = 1$  时计数的函数，因此在遇到对  $\gcd(i, j) = 1$  的计数问题中通常会用到它。

直接给出代码。

```
void InitMu() {
    mu[1] = 1;
```

```

for (int i = 2; i < N; i++) {
    if (!notPri[i]) pri[siz++] = i, mu[i] = -1;
    for (int j = 0; j < siz && i * pri[j] < N; j++) {
        int nxt = i * pri[j];
        notPri[nxt] = 1;
        if (i % pri[j])
            mu[nxt] = -mu[i];
        else {
            mu[nxt] = 0;
            break;
        }
    }
}
}

```

当出现平方因子就退出筛法保证了每个数只会被最小的因子筛去，因此时间复杂度线性 $\mu(i) = 0$  的情况是由最小因子筛掉的，而其他情况都是由  $\mu(i) = -\mu(j)$  得到的。

## 莫比乌斯反演

若函数  $f(n)$  与  $g(n)$  为数论函数，则

$$\sum f(n) = \sum_{d|n} g(d) \Leftrightarrow g(n) = \sum_{d|n} \mu \left( \frac{n}{d} \right) f(d)$$

这其实是说

$$g = f \times 1 \Leftrightarrow f = g \times \mu$$

一种理解的方法如下：

狄利克雷卷积中， $\mu$  的逆是  $\varepsilon$  即  $\varepsilon = 1 \times \mu$  这很容易理解：对  $(\mu \times 1)(n)$  作出贡献的仅有  $n$  的质因数的乘积和  $1$ 。

对于  $n$  的质因数，如果  $n$  有  $m \geq 1$  个质因数，那它就有  $\binom{m}{1}$  个“一个质因数的积” $\binom{m}{2}$  个“两个质因数的积……他们卷起来的和是

$$(-1) \cdot \binom{m}{1} + (-1)^2 \cdot \binom{m}{2} + \dots + (-1)^m \cdot \binom{m}{m} = [1 + (-1)]^m - 1$$

加上  $1$  的贡献，即为  $0$ 。所以只有当  $n=1$  的时候  $(\mu \times 1)(n)$  才为  $1$ ，故  $\varepsilon = 1 \times \mu$

莫比乌斯反演的另一种不太常用的形式是，若函数  $f(n)$  与  $g(n)$  为数论函数，则

$$\sum f(n) = \sum_{n|d} g(d) \Leftrightarrow g(n) = \sum_{n|d} \mu \left( \frac{d}{n} \right) f(d)$$

给出一些常用反演：

- $\varepsilon = \mu \times 1$
- $\varphi = \varphi \times 1 \Leftrightarrow \varphi = n \times \mu$

## 莫比乌斯反演与前缀、差分的联系

莫比乌斯反演实际上是一个高维前缀和与高维差分。

对  $n = \prod_{i=1}^m p_i^{k_i}$  则  $n$  代表了一个高维空间的点  $(k_1, k_2, \dots, k_m)$  且  $n$  的所有约数  $d$  代表的点与  $(k_1, k_2, \dots, k_m)$  的关系都是一个高维偏序关系。因此若有  $f = g \times 1$  则  $f$  实际是  $g$  的一个高维前缀和。类似地，与  $\mu$  的狄利克雷卷积可以将这个高维前缀和还原，因此  $\mu$  实际上是与之相对的高维差分。

利用这个性质，在计算类似  $g(n) = (f \times \mu)(n)$  时可以不需要枚举每个  $n$  的因数  $d$  而只用枚举其质因数  $p_i$  依次利用差分将每个  $p_i$  对应的维度降下来，可以略微降低复杂度。

## 应用

### 二维GCD计数前缀和

给定  $n, m, k$  求  $\sum_{i=1}^n \sum_{j=1}^m [\gcd(i,j) = k]$

不使用函数变换的方法

不难发现：

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m [(\gcd(i,j) = k)] &= \sum_{i=1}^n \sum_{j=1}^m \left( \sum_{d|k} \mu(d) \right) \\ &= \sum_{i=1}^n \sum_{d|k} \mu(d) \sum_{j=1}^{\lfloor n/d \rfloor} 1 \\ &= \sum_{d|k} \mu(d) \sum_{i=1}^{\lfloor n/d \rfloor} \sum_{j=1}^{\lfloor m/d \rfloor} 1 \end{aligned}$$

而  $\lfloor \frac{n}{d} \rfloor$  只有不超过  $\sqrt{n}$  种取值， $\lfloor \frac{m}{d} \rfloor$  只有不超过  $\sqrt{m}$  种取值，因此可以将  $[1, n]$  分成  $\sqrt{n} + \sqrt{m}$  块，每一块的  $\lfloor \frac{n}{d} \rfloor$  和  $\lfloor \frac{m}{d} \rfloor$  取值都不变，则我们预处理  $\mu$  后可以对一块区间进行  $\mathcal{O}(1)$  的统计，总时间复杂度为  $\mathcal{O}(\sqrt{n} + \sqrt{m})$

使用函数变换的方法

令  $f(k) = \sum_{i=1}^n \sum_{j=1}^m [(\gcd(i,j) = k)]$   $g(k) = \sum_{i=1}^n \sum_{j=1}^m [k \mid (i,j)]$  则  $f(k)$  就是我们要求的答案。很明显  $k \mid (i,j) \Leftrightarrow k \mid i \text{ 且 } k \mid j$  因此  $g(k) = \lfloor \frac{n}{k} \rfloor \lfloor \frac{m}{k} \rfloor$

发现  $g(k) = \sum_{d=1}^{\lfloor \frac{n}{k} \rfloor} f(d \times k)$  因此有：

$$\begin{aligned} f(k) &= \sum_{d=1}^{\lfloor \frac{n}{k} \rfloor} g(d \times k) \mu(d) \\ &= \sum_{d=1}^{\lfloor \frac{n}{k} \rfloor} \lfloor \frac{n}{dk} \rfloor \lfloor \frac{m}{dk} \rfloor \end{aligned}$$

令  $n' = \lfloor \frac{n}{k} \rfloor, m' = \lfloor \frac{m}{k} \rfloor$  则

$$\begin{aligned} f(k) = & \sum_{d=1}^{\lfloor n/k \rfloor} \left\lfloor \frac{n'}{d} \right\rfloor \cdot \mu(d) \\ & \left\lfloor \frac{m'}{d} \right\rfloor \end{aligned}$$

类似上面可以证明  $n', m'$  的取值个数，因此求解也是  $\mathcal{O}(\sqrt{n} + \sqrt{m})$  的。

好了，那求了一个区间后，怎么寻找下一个区间？假设我们当前区间开头为  $i$  并假设下一个区间为  $j$  则：

$$\begin{aligned} & \left\lfloor \frac{n'}{i} \right\rfloor \leq \left\lfloor \frac{n'}{j} \right\rfloor \Leftrightarrow \left\lfloor \frac{n'}{i} \right\rfloor < \left\lfloor \frac{n'}{j} \right\rfloor \Leftrightarrow \frac{n'}{i} < \frac{n'}{j} \Leftrightarrow j < i \end{aligned}$$

同理可得  $m'$  因此  $j = \min \left( \left\lfloor \frac{n'}{i} \right\rfloor, \left\lfloor \frac{m'}{i} \right\rfloor \right)$  这个技巧在很多莫比乌斯反演的题目都用得上。

## 求约数个数和

直接给出结论：

$$d(ij) = \sum_{x|ij} \sum_{y|x} [(x, y) = 1]$$

以下给出一个简单的证明：

上式显然先决定  $x$  的取值，再决定  $y$  的取值。对于一个因子  $p$  若  $p^a | i, p^b | j$  且  $p^{a+b} | ij$  则

- $p^0 | x$  表示  $y$  可以任意选  $p^1, \dots, p^{a+b}$  等因子，分别映射到因数  $d | mid p^{a+1}, d | mid p^{a+2}, \dots, d | mid p^{a+b}$
- $p^1 | x, p^2 | x, \dots, p^a | x$  表示  $x$  可以任意选  $p^1, \dots, p^a$  等因子，分别映射到因数  $d | mid p^1, d | mid p^2, \dots, d | mid p^a$
- $p^0 | x$  且  $q^0 | y$  等因子，映射到因数  $d | mid p^0$

综上，因子  $p^0, p^1, \dots, p^{a+b}$  都能被唯一地表示出来且一一对应（双射），因此等式成立。

然后还有个推广的神奇大结论：

$$\begin{aligned} & \sum_{x_1}^y \sum_{x_2}^y \cdots \sum_{x_k}^y d(x_1 x_2 \cdots x_k) = \\ & \sum_{x_1}^y \sum_{x_2}^y \cdots \sum_{x_k}^y \prod_{i=1}^k \left\lfloor \frac{y}{x_i} \right\rfloor \end{aligned}$$

太神奇，[证明](#) 需要二重数学归纳，略过。

## 练习

莫比乌斯的题目通常能转化为  $(i, j) = 1$  的计数问题，而转化为计数问题我们就容易通过分块求解了。

## POI2007 Zap

二维 GCD 计数前缀和。

## HAOI2011 Problem b

POI2007 Zap 的加强版，容斥原理加加减减就好了。

## BZOJ2820 YY的GCD

仍然是二维 GCD 计数前缀和，不过需要  $(i,j)$  为质数。只要预处理质数的  $\mu$  前缀和就好了。

## SDOI2008 仪仗队

不被挡住即行列  $(i,j)=1$  从 \$0\$ 标号），因此答案为  $(\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} [(i,j)=1]) + 2 \lfloor \frac{n}{2} \rfloor$  个是  $(0,1)$ 、 $(1,0)$ 。最终化为  $(\sum_{g=1}^n \mu(g) \lfloor \frac{n}{g} \rfloor)^2 + 2$  分块求解。

## SDOI2015 约数个数和

是道好题，然而需要结论。

令  $n' = \frac{n}{g}$   $m' = \frac{m}{g}$  则

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^m d(ij) = \sum_{i=1}^n \sum_{j=1}^m [d(ij) = 1] \\ &= \sum_{i=1}^n \sum_{j=1}^m \left\{ \lfloor \frac{ni}{g} \rfloor \right\} \left\{ \lfloor \frac{mj}{g} \rfloor \right\} \sum_{g \mid (i,j)} \mu(g) \\ &= \sum_{g=1}^n \mu(g) \sum_{i=1}^{\lfloor n/g \rfloor} \sum_{j=1}^{\lfloor m/g \rfloor} \left\{ \lfloor \frac{ni}{g} \rfloor \right\} \left\{ \lfloor \frac{mj}{g} \rfloor \right\} \\ &= \sum_{g=1}^n \mu(g) \sum_{i=1}^{\lfloor n/g \rfloor} \sum_{j=1}^{\lfloor m/g \rfloor} \left( \lfloor \frac{ni}{g} \rfloor - \lfloor \frac{(i-1)n}{g} \rfloor \right) \left( \lfloor \frac{mj}{g} \rfloor - \lfloor \frac{(j-1)m}{g} \rfloor \right) \end{aligned}$$

然后就可以预处理  $f(n) = \sum_{i=1}^n \left\{ \lfloor \frac{ni}{g} \rfloor \right\}$  的值，每次询问就可以分块解决。之所以要预处理  $f(n)$  是因为在倒数第二步时如果采用直接计算  $\sum_{i=1}^{\lfloor n/g \rfloor} \sum_{j=1}^{\lfloor m/g \rfloor} \left( \lfloor \frac{ni}{g} \rfloor - \lfloor \frac{(i-1)n}{g} \rfloor \right) \left( \lfloor \frac{mj}{g} \rfloor - \lfloor \frac{(j-1)m}{g} \rfloor \right)$  开销是很大的。但如果我们将  $f(n)$  预处理，就能做到  $\mathcal{O}(1)$  计算。

预处理时间复杂度  $\mathcal{O}(n\sqrt{n})$  单次询问时间复杂度  $\mathcal{O}(\sqrt{n})$

## HNMT2015#5 Lucas的数论

发现是 SDOI2015 约数个数和的单询问加强版本，上面对  $\mu$  前缀和的  $\mathcal{O}(n)$  时间复杂度已经不能满足我们了，因此我们需要用杜教筛求出  $\mu(n)$  前缀和，在  $\mathcal{O}(n^{2/3})$  时

间内完成计算。

## LOJ6627 等比数列三角形

比较有意思的莫比乌斯反演题。

枚举比值  $k=\frac{p}{q} \geq 1$  中的  $p, q$  再枚举最短边  $x$  使得  $x+kx > k^2x$  可以得到约束  $\Phi = \frac{1+\sqrt{5}}{2}$  即  $p < \Phi q$  同时  $x$  需要为  $q^2$  的整数倍，故  $q \leq \sqrt{n}$  设  $x = iq^2$  式子化为：

$$\begin{aligned} & \sum_{q=1}^{\sqrt{n}} \sum_{(p,q)=1}^{\infty} \sum_{i=1}^{+\infty} [ip^2 \leq n] \\ &= \sum_{q=1}^{\sqrt{n}} \sum_{(p,q)=1}^{\infty} \left\lfloor \frac{n}{q^2} \right\rfloor \end{aligned}$$

发现可以将  $p$  前提，故找出  $p$  的范围。由于  $xk^2 = ip^2 \leq n$  原式进一步化为：

$$\sum_{p=1}^{\sqrt{n}} \left\lfloor \frac{n}{p^2} \right\rfloor = \sum_{p=1}^{\lceil p \rceil} \sum_{(i,p)=1}^{\infty} \sum_{d \mid p} \mu(d) \left\lfloor \frac{n}{d^2} \right\rfloor$$

此时考虑简化对  $q$  求和部分。我们发现这部分就是  $\varphi(p) - \sum_{i=1}^{\lceil p \rceil} \sum_{(i,p)=1}^{\infty} \sum_{d \mid p} \mu(d) \left\lfloor \frac{n}{d^2} \right\rfloor$  枚举  $p$  以及  $p$  的每个因数  $d$  由于  $\sim n$  约数个数和是  $O(n \log n)$ ，故复杂度为  $O(\sqrt{n} \log n)$

[这里](#) 提供了一个  $\varphi(n) \log \log \sqrt{n}$  的做法，该做法基于莫比乌斯反演的前缀和差分意义，使得计算  $g(n) = (f * \mu)(n)$  时不需要枚举每个  $n$  的因数  $d$  而只用枚举质因数  $p_i$  使复杂度略微下降。

## 总结

莫比乌斯反演基本上离不开 GCD 和两个累和符号，而且通常通过将式子化为  $\varepsilon(n)$  的形式，进而反演成  $\mu(n)$  并提出相关变量的形式，简化式子进行计算。求解一般通过数论分块和预处理  $\mu(n)$  前缀和的方式在  $O(\sqrt{n})$  时间内求和。

- $\varepsilon = \mu * 1$
- 当待分块函数（如  $\mu$ ）可以单独提出预处理时，可以通过此降低时间复杂度。
- 若多次询问中，分块区域下含有 GCD 的枚举值  $g$  和  $i$  或  $j$  之一，可以通过更换枚举变量改为枚举  $ig$  或  $jk$  的值，再枚举  $g$  加速。（说法很意识流，详见[莫比乌斯反演简要笔记 - GCD的幂](#)）
- 积性函数有时不好证明，可以打表观察。重点观察幂和质数的值。

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

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i\\_dont\\_know\\_png:nikkukun:mobius\\_inversion&rev=1589273354](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:nikkukun:mobius_inversion&rev=1589273354)

Last update: 2020/05/12 16:49

