

数论函数

Number Theory Functions

积性函数 Multiplicable function

Definition:

- $\forall x, y \in \mathbb{N}_+$, $\gcd(x, y) = 1$, 都有 $f(xy) = f(x)f(y)$, 则称 $f(x)$ 是积性函数。
- $\forall x, y \in \mathbb{N}_+$, 都有 $f(xy) = f(x)f(y)$, 则称 $f(x)$ 是完全积性函数。

Properties: 若 $f(x), g(x)$ 均为积性函数, 则下列函数也是积性函数:

$$\begin{aligned}h(x) &= f(x^p) \\h(x) &= f^p(x) \\h(x) &= f(x)g(x) \\h(x) &= \sum_{d|x} f(d)g\left(\frac{x}{d}\right)\end{aligned}$$

Examples:

- 单位函数: $\epsilon(n) = [n = 1]$
- 恒等函数: $\text{id}_k(n) = n^k$
 - $k = 1$ 时, $\text{id}(n) = n$ 即标号函数
- 常数函数: $1(n) = 1$
- 除数函数: $\sigma_k(n) = \sum_{d|n} d^k$
 - $k = 0$ 时, $\sigma_0(n) = d(n) = \sum_{d|n} 1$ 即约数个数函数
 - $k = 1$ 时, $\sigma(n) = \sum_{d|n} d$ 即约数和函数
- 欧拉函数: $\varphi(n) = \sum_{i=1}^n [\gcd(i, n) = 1]$
- 莫比乌斯函数: $\mu(n) = \begin{cases} 1 & n = 1 \\ 0 & \exists d > 1: d^2 | n, \text{ 其中, } \omega(n) \text{ 表示 } n \text{ 的不同质因数个数} \\ (-1)^{\omega(n)} & \text{otherwise} \end{cases}$

欧拉函数 Euler's totient function

Definition: $\varphi(n)$ 表示小于等于 n 的与 n 互质的数的个数。

Theorem: 设 $n = \prod_{i=1}^k p_i^{r_i}$ (由唯一分解定理给出), 则:

$$\varphi(n) = n \cdot \prod_{i=1}^k \left(1 - \frac{1}{p_i}\right)$$

Intuitive proof: 对于 p_i 来说, p_i 及其倍数是 n 的因数, 占比 $\frac{1}{p_i}$, 则非 p_i 或其倍数者占比 $\left(1 - \frac{1}{p_i}\right)$; 故既非 p_1 倍数、又非 p_2 倍数.....又非 p_k 倍数的数占比 $\prod_{i=1}^k \left(1 - \frac{1}{p_i}\right)$, 故 $\varphi(n) = n \cdot \prod_{i=1}^k \left(1 - \frac{1}{p_i}\right)$.

Properties:

- 设 p 是素数, $\varphi(p) = p - 1$. (由定义显然)
- 设 p 是素数, $n = p^k$, 则 $\varphi(n) = p^k - p^{k-1}$.

- 积性：若 $(a, b) = 1$ ，则 $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.
- 对于素数 p ： $\varphi(n \cdot p) = \begin{cases} \varphi(n) \cdot p & p \mid n \\ \varphi(n) \cdot (p - 1) & p \nmid n \end{cases}$.
- 小于等于 n 与 n 互质的数的总和为 $\varphi(n) \cdot \frac{n}{2}$.

证明：若 m 与 n 互质，则 $n - m$ 也与 n 互质，故所有小于等于 n 与 n 互质的数的平均数为 $\frac{n}{2}$ ，故总和为 $\varphi(n) \cdot \frac{n}{2}$.

- $n = \sum_{d \mid n} \varphi(d)$.

证明：设 $f(x)$ 表示 $\gcd(k, n) = x$ ($k \leq n$) 的数的个数，则 $n = \sum_{i=1}^n f(i)$ 。又 $\gcd(k, n) = x \iff \gcd\left(\frac{k}{x}, \frac{n}{x}\right) = 1$ ，所以 $f(x) = \varphi\left(\frac{n}{x}\right)$ ，故 $n = \sum_{i=1}^n \varphi\left(\frac{n}{i}\right) = \sum_{d \mid n} \varphi(d)$.

Code（应用公式求出单个 $\varphi(n)$ ）：

Complexity: $O(\sqrt{n})$

```
1  int getPhi(int n){
2      int res = n;
3      for(int i = 2; i * i <= n; i++){
4          if(n % i == 0){
5              res = res / i * (i - 1);
6              while(n % i == 0)
7                  n /= i;
8          }
9      }
10     if(n > 1)    res = res / n * (n - 1);
11     return res;
12 }
```

Code（线性筛）：

Complexity: $O(n)$

```
1  int phi[N], pList[N], pID;
2  bool notP[N];
3  void Euler(int n){
4      notP[0] = notP[1] = 1;
5      phi[1] = 1;
6      for(int i = 1; i <= n; i++){
7          if(notP[i] == 0){
8              pList[++pID] = i;
9              phi[i] = i - 1;
10         }
11         for(int j = 1; j <= pID; j++){
12             if(1ll * i * pList[j] > n) break;
13             notP[i * pList[j]] = 1;
14             if(i % pList[j] == 0){
15                 phi[i * pList[j]] = phi[i] * pList[j];
16                 break;
17             }
18             else    phi[i * pList[j]] = phi[i] * (pList[j] - 1);
19         }
20     }
21 }
```

莫比乌斯函数 Möbius function

Definition: $\mu(n) = \begin{cases} 1 & n = 1 \\ 0 & \exists d > 1 : d^2 \mid n \\ (-1)^k & n = p_1 p_2 \cdots p_k \end{cases}$

Properties:

- $\sum_{d|n} \mu(d) = \begin{cases} 1, & n = 1 \\ 0, & n > 1 \end{cases} = \epsilon(n).$

证明: 设 $n = \prod_{i=1}^k p_i^{r_i}$, 则 $\sum_{d|n} \mu(d) = \sum_{i=0}^k \binom{k}{i} (-1)^i = (1 + (-1))^k = \begin{cases} 1, & n = 1 \\ 0, & n > 1 \end{cases}.$

推论: $[\gcd(i, j) = 1] \iff \sum_{d|\gcd(i, j)} \mu(d)$

- $\sum_{d|n} \frac{\mu(d)}{d} = \frac{\varphi(n)}{n}.$
- 积性: 若 $(a, b) = 1$, 则 $\mu(a \times b) = \mu(a) \times \mu(b).$ (由定义易证)

Code (线性筛):

Complexity: $O(n)$

```

1  int mu[N], pList[N], pID;
2  bool notP[N];
3  void Euler(int n){
4      notP[0] = notP[1] = 1;
5      mu[1] = 1;
6      for(int i = 1; i <= n; i++){
7          if(notP[i] == 0){
8              pList[++pID] = i;
9              mu[i] = -1;
10         }
11         for(int j = 1; j <= pID; j++){
12             if(1ll * i * pList[j] > n) break;
13             notP[i * pList[j]] = 1;
14             if(i % pList[j] == 0){
15                 mu[i * pList[j]] = 0;
16                 break;
17             }
18             else mu[i * pList[j]] = -mu[i];
19         }
20     }
21 }
```

狄利克雷卷积 Dirichlet Convolution

Definition: 两个数论函数 f, g 的 Dirichlet 卷积为:

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

或

$$(f * g)(n) = \sum_{ab=n} f(a)g(b)$$

Properties:

- 满足交换律:

$$f * g = g * f$$

易证。

- 满足结合律：

$$(f * g) * h = f * (g * h)$$

从外往内计算就容易知道，多个函数的 **Dirichlet** 卷积就是遍历自变量乘积为 n 的所有可能的函数值相乘的和。

- 满足分配律：

$$(f + g) * h = f * h + g * h$$

易证。

- $\epsilon(n)$ 是 **Dirichlet** 卷积的单位元，即任何函数卷 $\epsilon(n)$ 均为它本身。
- 积性函数的 **Dirichlet** 卷积仍为积性函数。

证明：令 $h = f * g$ 。设 $(n, m) = 1$ ，则 $h(n) \times h(m) = \sum_{d|n} f(d)g\left(\frac{n}{d}\right) \times \sum_{k|m} f(k)g\left(\frac{m}{k}\right)$ ，考虑前者的第 i 项和后者的第 j 项

相乘得出的项： $f(d_i)g\left(\frac{n}{d_i}\right) \times f(k_j)g\left(\frac{m}{k_j}\right) = f(d_i k_j)g\left(\frac{nm}{d_i k_j}\right)$ ，当 d_i 遍历所有 n 的因子、 k_j 遍历所有 m 的因子时，由于 n, m 互质， $d_i k_j$ 遍历了所有 nm 的因子。故 $h(n) \times h(m) = \sum_{d|nm} f(d)g\left(\frac{nm}{d}\right) = h(n \times m)$ 。

- 逆元：对于 $f(1) \neq 0$ 的函数 $f(n)$ ， $\exists g(n)$ 使得 $f(n) * g(n) = \epsilon(n)$ 。

事实上，有：

$$g(n) = \frac{1}{f(n)} \left([n=1] - \sum_{d|n, d>1} f(d)g\left(\frac{n}{d}\right) \right)$$

或：

$$g(n) = \begin{cases} \frac{1}{f(n)} & n = 1 \\ -\frac{1}{f(n)} \sum_{d|n, d>1} f(d)g\left(\frac{n}{d}\right) & n > 1 \end{cases}$$

注意这是一个递归定义。

注意：积性函数必有逆（因为积性函数一定有 $f(1) = 1 \neq 0$ ），且逆元也是积性函数。

Examples:

$$\epsilon = \mu * 1 \iff \epsilon(n) = \sum_{d|n} \mu(d)$$

$$d = 1 * 1 \iff d(n) = \sum_{d|n} 1$$

$$\sigma = \text{id} * 1 \iff \sigma(n) = \sum_{d|n} d$$

$$\varphi = \mu * \text{id} \iff \varphi(n) = \sum_{d|n} \mu(d) \frac{n}{d} = \sum_{d|n} d \cdot \mu\left(\frac{n}{d}\right)$$

$$\text{id} = \varphi * 1 \iff \text{id}(n) = n = \sum_{d|n} \varphi(d)$$

其他性质 Properties

除上述卷积的例子以外，其他有关积性函数的性质。

$$\begin{aligned}d(i \cdot j) &= \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1] \\ \sigma(i \cdot j) &= \sum_{a|i} \sum_{b|j} [\gcd(a, b) = 1] \frac{a \cdot b}{b} \\ \varphi(i \cdot j) &= \varphi(i) \varphi(j) \frac{\gcd(i, j)}{\varphi(\gcd(i, j))}\end{aligned}$$

莫比乌斯反演 Möbius Inversion

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

或写成 **Dirichlet** 卷积形式：

$$f = g * 1 \iff g = f * \mu$$

Proof: 运用卷积，因为 $\mu(n)$ 是 $1(n)$ 的逆元，即 $\mu * 1 = \epsilon$ ，所以有：

$$f = g * 1 \iff f * \mu = g * 1 * \mu \iff f * \mu = g$$

常用技巧#1：数论分块

假设我们要求含有 $\left\lfloor \frac{n}{d} \right\rfloor$ 的和式（例如 $\sum_{d=1}^n \left\lfloor \frac{n}{d} \right\rfloor$ ），发现其中有很多项值其实是相同的。例如： $\left\lfloor \frac{60}{21} \right\rfloor = \left\lfloor \frac{60}{22} \right\rfloor = \dots = \left\lfloor \frac{60}{30} \right\rfloor$ ，所以我们考虑能否把它们放在一起算。换句话说，我们要找到最大的 r ，使 $\left\lfloor \frac{n}{r} \right\rfloor = \left\lfloor \frac{n}{l} \right\rfloor$ 。可以证明： $r = \left\lfloor \frac{n}{\left\lfloor \frac{n}{l} \right\rfloor} \right\rfloor$ 。

首先， $\left\lfloor \frac{n}{l} \right\rfloor = \left\lfloor \frac{n}{r} \right\rfloor \leq \frac{n}{r}$ ，于是 $r \leq \frac{n}{\left\lfloor \frac{n}{l} \right\rfloor}$ ，故 $r = \left\lfloor \frac{n}{\left\lfloor \frac{n}{l} \right\rfloor} \right\rfloor$ 。其次， $r \leq n$ 显然，需要证明 $r \geq l$ ：
 $r = \left\lfloor \frac{n}{\left\lfloor \frac{n}{l} \right\rfloor} \right\rfloor \geq \left\lfloor \frac{n}{\frac{n}{l}} \right\rfloor = \lfloor l \rfloor = l$ 。证毕。

于是乎，我们把 $[l, r]$ 视为一块，分块求和。

可以证明：块数不会超过 $2\sqrt{n}$ 。

当 $i < \sqrt{n}$ 时， $\left\lfloor \frac{n}{i} \right\rfloor$ 最多一个数一块，最多 \sqrt{n} 块；当 $i \geq \sqrt{n}$ 时， $\left(\frac{n}{2}, n\right]$ 是一块， $\left(\frac{n}{3}, \frac{n}{2}\right]$ 是一块，……， $\left(\frac{n}{\sqrt{n}+1}, \frac{n}{\sqrt{n}}\right]$ 是一块，最多 \sqrt{n} 块。证毕。

Complexity: $O(\sqrt{n})$

Code:

以 $\sum_{i=1}^n \left\lfloor \frac{k}{i} \right\rfloor$... 为例：

```
1 for(long long l = 1, r; l <= n; l = r + 1){
2     if(k / l == 0) r = n;
3     else r = min(n, k / (k / l));
4     ... // calculate the ans
5 }
```

常用技巧#2：提取公因数

两个和式情形：

$$\sum_{i=1}^n \sum_{d|i} f(d, i) \stackrel{i \rightarrow di}{=} \sum_{d=1}^n \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} f(d, di)$$

即考虑 d 对答案的贡献：凡是 d 的倍数都会对答案有贡献。

三个和式情形：

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{d|\gcd(i,j)} f(d, i, j) \stackrel{i \rightarrow di, j \rightarrow dj}{=} \sum_{d=1}^{\min(n,m)} \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} \sum_{j=1}^{\lfloor \frac{m}{d} \rfloor} f(d, di, dj)$$

即考虑 d 对答案的贡献：凡 d 是 i, j 的公因子就会对答案有贡献。

更多和式可类似拓展。

常用技巧#3：技巧2的逆过程

$$\sum_{p=1}^n \sum_{d=1}^{\lfloor \frac{n}{p} \rfloor} f(p, d) = \sum_{1 \leq p \cdot d \leq n} f(p, d) \stackrel{T=p \cdot d}{=} \sum_{T=1}^n \sum_{p|T} f\left(p, \frac{T}{p}\right)$$

常用技巧#4：一种预处理

$$g(n) = \sum_{d|n} f(d, n)$$

上式的预处理方式：枚举 d ，将 $d | n$ 的 $g(n)$ 加上贡献 $f(d, n)$ 。

复杂度是 $O(nH(n))$ ，大概是 $O(n \lg n)$ 的。