初等数论

同余、欧拉定理、裴蜀定理等

郑欣

2024 年 7 月 29 日

对于两个整数 d, a $(d > 0)$，存在两个唯一的整数 q, r，满足

$$a = q \cdot d + r \quad (0 \le r < d)$$

称 q, r 分别为 $a \div d$ 的商和余数，记 $q = \lfloor a/d \rfloor$, $r = a \bmod d$。

当 $r = 0$ 时，称 a 整除 d，记作 $d \mid a$。

Code

```
// q = a / d, r = a % d 只在 a ≥ 0 时成立
// 但是 (a / d) * d + a % d = a 始终成立
r = a % d, r += r < 0 ? d : 0;
q = (a - r) / d;
```

约数与质数

对于正整数 d, a，若 $d \mid a$，则称 d 是 a 的约数。

如果 $a \geq 2$，且 a 没有 1 和 a 以外的约数，则称 a 是质数。

一些事实：

- 质数有无穷多个；
- $\pi(n) \to n/\ln n$（$\pi(n)$ 表示 n 以内的质数数量）；
- $p_n = \Theta(n \log n)$（p_n 表示第 n 个质数）；
- $\sum_{i=1}^{n} 1/p_i = \Theta(\log\log n)$（$\sum_{i=1}^{n} 1/i = \Theta(\log n)$）。

算术基本定理

算术基本定理（唯一分解定理）

任意一个正整数 n 都可以表示为有限个质数的乘积，即存在 $n_1, n_2, \ldots, n_k \geq 0$，使得

$$n = p_1^{n_1} p_2^{n_2} \ldots p_k^{n_k},$$

其中 p_i 表示第 i 个质数。

公约数与公倍数

- **公约数**:

  对于正整数 a, b，若 $d \mid a$ 且 $d \mid b$，则称 d 是 a, b 的公约数。

  对于其中最大的 d，称 d 为最大公约数，记作 $d = \gcd(a, b)$。

  约定 $\gcd(0, a) = a$。

- **公倍数**:

  对于正整数 a, b，若 $a \mid d$ 且 $b \mid d$，则称 d 是 a, b 的公倍数。

  对于其中最小的 d，称 d 为最小公倍数，记作 $d = \operatorname{lcm}(a, b)$。

令 $d = p_1^{d_1} p_2^{d_2} \dots p_k^{d_k}$，$a = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$，则 $d \mid a$ 当且仅当对任意 $1 \le i \le k$ 有 $d_i \le a_i$。

类似地，$d \mid b$ 当且仅当对任意 $1 \le i \le k$ 有 $d_i \le b_i$。

因此 d 同时满足 $d \mid a$ 和 $d \mid b$ 当且仅当 $d_i \le \min\{a_i, b_i\}$，所以

$$\gcd(a, b) = p_1^{\min\{a_1, b_1\}} p_2^{\min\{a_2, b_2\}} \dots p_k^{\min\{a_k, b_k\}}$$

同理，

$$\operatorname{lcm}(a, b) = p_1^{\max\{a_1, b_1\}} p_2^{\max\{a_2, b_2\}} \dots p_k^{\max\{a_k, b_k\}}$$

一些性质：

- $\gcd(a,b)\cdot \operatorname{lcm}(a,b)=ab$；
- $\gcd(ac,bc)=c\cdot\gcd(a,b)$；
- 若 $\gcd(a,b)=1$，则 $a\mid c,\ b\mid c\ \Rightarrow\ ab\mid c$；
- 若 $\gcd(a,b)=1$，则 $a\mid bc\ \Rightarrow\ a\mid c$；
- 若 $\gcd(a,b)=1$，则 $\gcd(a,bc)=\gcd(a,c)$；

称 a 和 b 互质，若 $\gcd(a,b)=1$。

特别地，1 与所有正整数互质。

例题 1

Luogu P1029　最大公约数和最小公倍数问题（加强）

给定 a, b，求满足 $\gcd(x, y) = a$ 且 $\operatorname{lcm}(x, y) = b$ 的 (x, y) 个数。

范围：$a, b \le 10^{12}$

假设 a, b 分别有 a_i, b_i 个质因子 p_i $(a_i \le b_i)$，则我们需要 x, y 中的质因子数量满足 $\min\{x_i, y_i\} = a_i$ 且 $\max\{x_i, y_i\} = b_i$。因此当 $a_i < b_i$ 时有两组解，$a_i = b_i$ 时有唯一解。

每个质因子结果相互独立，所以把每个质因子结果乘起来即可。

复杂度 $O(\sqrt{\max\{a, b\}})$。

更相减损术

更相减损术：当 $a \geq b$，$\gcd(a,b) = \gcd(a-b,b)$。

可以用来求最大公约数，但是复杂度最坏 $O(a)$。

推论：

- 对任意整数 q，$\gcd(a,b) = \gcd(a+qb,b)$；
- $\gcd(a,b) = \gcd(a \bmod b, b)$ $\left(a \bmod b = a - \lfloor a/b \rfloor \cdot b\right)$。

欧几里得算法（辗转相除法）：$\gcd(a, b) = \gcd(b, a \bmod b)$。

复杂度 $O(\log \min\{a, b\})$：当 $a \geq b$，$a \bmod b < a/2$。

可以直接用库函数 `std::__gcd`（C++ 17 后可以用 `std::gcd`）实现。

Code

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
```

观察：欧几里得算法中，每一步函数的两个参数都是 a 和 b 的线性组合。

裴蜀定理

对于任意整数 a, b（a, b 不全为 0）：

- 对任意整数 x, y，有 $\gcd(a, b) \mid ax + by$；
- 存在整数 x, y，使得 $\gcd(a, b) = ax + by$。

推论：若 $ax + by > 0$，则 $ax + by$ 的最小值为 $\gcd(a, b)$。

如何求出满足 $ax + by = \gcd(a, b)$ 的 x, y:

- 如果 $b = 0$，那么 $x = 1$, $y = 0$。
- 否则令 $a' = b$, $b' = a \bmod b$。假设

$$a'x' + b'y' = \gcd(a, b),$$

则

$$bx' + \left(a - \left\lfloor \frac{a}{b} \right\rfloor b\right) y' = \gcd(a, b),$$

即

$$ay' + b\left(x' - \left\lfloor \frac{a}{b} \right\rfloor y'\right) = \gcd(a, b).$$

因此令 $x = y'$, $y = x' - \lfloor a/b \rfloor y'$ 即可。

拓展欧几里得算法

$$ax + by = \gcd(a, b) \quad \Rightarrow \quad bx' + (a \bmod b)\, y' = \gcd(a, b):$$

$$x = y', \quad y = x' - \left\lfloor \frac{a}{b} \right\rfloor y'$$

Code

```
typedef long long ll;

// ax + by = d = gcd(a, b)
void exgcd(ll a, ll b, ll &d, ll &x, ll &y) {
    if (!b) d = a, x = 1, y = 0;
    else exgcd(b, a % b, d, y, x), y -= x * (a / b);
}
```

算法只会求出一组解 (x_0, y_0)。当 $b \neq 0$ 时这个解满足 $|x_0| \leq b/\gcd(a, b)$，$|y_0| \leq a/\gcd(a, b)$。

方程的所有解：

$$x = x_0 + k\frac{b}{\gcd(a, b)}, \quad y = y_0 - k\frac{a}{\gcd(a, b)} \quad (k \in \mathbb{N})$$

例题 3

CF 1427E　Xum

一个序列，序列一开始有一个奇数 x。每次操作可以选两个数（可以相同）a, b，然后把 $a+b$ 或 $a \oplus b$ 加入序列。构造一种方案使得序列里出现 1。

范围：$3 \le x < 10^6$

只要出现两个互质的数 x, y，就可以找到 $a, b \ge 0$ 使得 $ax - by = 1$。

ax 和 by 可以用类似快速幂的方式求出；如果 ax 是奇数，by 是偶数，那么 $ax \oplus by = 1$。

问题转化为怎么构造两个互质的数。

例题 3

CF 1427E　Xum

一个序列，序列一开始有一个奇数 x。每次操作可以选两个数（可以相同）a, b，然后把 $a+b$ 或 $a \oplus b$ 加入序列。构造一种方案使得序列里出现 1。

范围：$3 \le x < 10^6$

只要出现两个互质的数 x, y，就可以找到 $a, b \ge 0$ 使得 $ax - by = 1$。

ax 和 by 可以用类似快速幂的方式求出；如果 ax 是奇数，by 是偶数，那么 $ax \oplus by = 1$。

问题转化为怎么构造两个互质的数。

假设 x 有 $n+1$ 个二进制位（$n \ge 1$），则 $2^n x \oplus x = (2^n+1)x - 2^{n+1}$。因为 $2 \nmid x$，所以

$$\gcd(2^n x \oplus x,\ x) = \gcd(-2^{n+1}, x) = 1.$$

若 $m \mid a-b$，则称 a 和 b 在模 m 意义下同余，记作 $a \equiv b \pmod m$。

性质：

- $\equiv$ 是等价关系；
- 对于任意 a，存在 $0 \le b < m$，使得 $a \equiv b \pmod m$；
- 若 $d \mid \gcd(a,b,m)$，则 $a \equiv b \pmod m \;\Rightarrow\; a/d \equiv b/d \pmod{m/d}$；
- 若 $a \equiv a'$，$b \equiv b' \pmod m$，则 $a+b \equiv a'+b'$，$ab = a'b' \pmod m$。

在 $\bmod m$ 意义下是否能进行类似除法的操作？

即定义 $x \equiv d/a \pmod m$，其中 x 满足 $ax \equiv d \pmod m$。

同余方程

$$ax \equiv d \pmod{m} \quad \Leftrightarrow \quad ax + my = d$$

用 exgcd 求出满足 $ax' + my' = \gcd(a, m)$ 的 x', y'：

- 若 $\gcd(a, m) \nmid d$，则方程无解。
- 若 $\gcd(a, m) \mid d$，则
$$x = \frac{d}{\gcd(a, m)} x', \quad y = \frac{d}{\gcd(a, m)} y'$$
是 $ax + my = d$ 的一组解。

可以顺手切掉 Luogu P1082。

逆元

当 $d=1$，即 $ax \equiv 1 \pmod m$ 时，称 x 为 a 在模 m 意义下的乘法逆元，记 $x = a^{-1}$。

Code (逆元 1)

```
ll inv(ll a, ll m) {
    ll d, x, y;
    exgcd(a, m, d, x, y);
    return (x + m) % m;
}
```

性质：

- $a \cdot a^{-1} \equiv 1 \pmod m$；
- a^{-1} 存在当且仅当 $\gcd(a, m) = 1$。

$d/a \bmod m$ 可以看作 $da^{-1} \bmod m$，因为 $da^{-1} \cdot a \equiv d(a^{-1}a) \equiv d \pmod m$。

逆元

特别地，当 m 是质数，对所有 $1 \le a < m$ 都有 $\gcd(a, m) = 1$，所以 a^{-1} 一定存在（除了 0，就像 0 不能做分母）。因此 a^{-1} 可以看作模意义下的除法运算。

很多涉及到有理数运算的题目都会用输出 $\bmod p$ 的结果替代输出实数值（其中 p 是一个质数），以避免精度问题。

常见的 p 有 $998\,244\,353$，$10^9 + 7$ 等。

预处理逆元

如果 p 是质数且需要多次用比较小的数的逆元，可以递推预处理逆元。这样求逆元的时候只需要 $O(1)$ 查询：

- $1^{-1} = 1$；
- 当 $i \geq 2$，由于 $p \bmod i = p - \lfloor p/i \rfloor \cdot i$，两边乘 i 对 p 的逆元 i^{-1} 有

$$(p \bmod i)\, i^{-1} \equiv p i^{-1} - \left\lfloor \frac{p}{i} \right\rfloor \cdot i\, i^{-1} \pmod p,$$

即 $(p \bmod i)\, i^{-1} \equiv -\lfloor p/i \rfloor \pmod p$。再两边乘 $(p \bmod i)^{-1}$ 有

$$i^{-1} \equiv -\left\lfloor \frac{p}{i} \right\rfloor \cdot (p \bmod i)^{-1} \pmod p$$

Code (逆元 2)

```
inv[1] = 1;
for (int i = 2; i <= n; ++i)
    inv[i] = (p - p / i) * inv[p % i] % p;
```

预处理逆元

如果需要预处理阶乘的逆元，用 f_i 存放 $i!$，g_i 存放 $(i!)^{-1}$：

- 先预处理 $f_i = i!$，即 $f_0 = 1, f_i = i \cdot f_{i-1}$；
- 求 $g_n = f_n^{-1}$；
- 注意到 $g_{i-1} = ((i-1)!)^{-1} = i \cdot (i!)^{-1}$，可以倒着递推：$g_{i-1} = i \cdot g_i$。

常用于 $O(1)$ 求组合数：

$$\binom{m}{n} = \frac{m!}{n!(m-n)!} \equiv m! \cdot (n!)^{-1} \cdot ((m-n)!)^{-1} \pmod p$$

如果 p 不是质数，或者 m, n 比较小的情况可以用杨辉三角递推。

威尔逊定理

对于任意质数 p，$(p-1)! \equiv -1 \pmod{p}$。

- $a = a^{-1}$ 当且仅当 $a = 1$ 或 $a = -1$。
- $(a^{-1})^{-1} = a$。

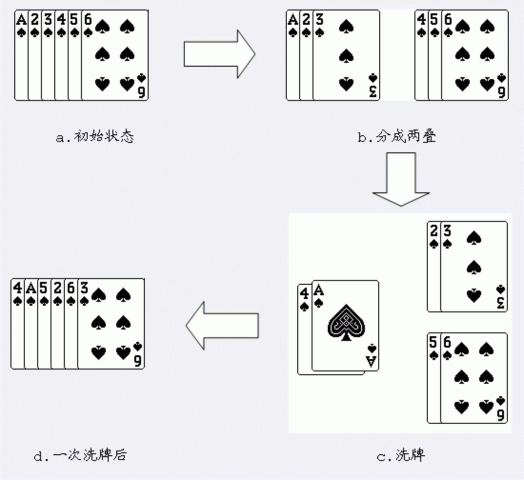
所以 $2 \sim p-2$ 中的所有数可以 a 与 a^{-1} 匹配相乘，全部乘起来是 1，即 $(p-2)! \equiv 1 \pmod{p}$。

所以 $(p-1)! \equiv (p-2)! \cdot (p-1) \equiv -1 \pmod{p}$。

例题 1

AHOI2005　洗牌

n 张牌，进行 m 次洗牌操作，问最后第 ℓ 张牌是什么（$n, m \leq 10^{10}$）。



例题 1

AHOI2005　洗牌

n 张牌，进行 m 次洗牌操作，问最后第 ℓ 张牌是什么。

范围：$n, m \le 10^{10}$

一次操作就是把位置 x 上的牌移到了 $2x \bmod (n+1)$，因此 m 次操作后就是 $2^m x \bmod (n+1)$。

反之要求 ℓ 上的牌解方程 $2^m x \equiv \ell \pmod{n+1}$ 即可。

模 m 下的完全剩余系：对 m 取模后能够恰好取遍 $0 \sim m-1$ 的 m 个数。

性质：在模 m 的完全剩余系中，

- 对每个数 $+a$ 后还是完全剩余系，即 $(i+a) \bmod m$ 取遍 $0 \sim m-1$。
- 若 $\gcd(a,m)=1$，则对每个数 $\times a$ 后还是完全剩余系，即 $(i \cdot a) \bmod m$ 取遍 $0 \sim m-1$。

费马小定理

对于任意质数 p 与 $1 \le a < p$，$a^{p-1} \equiv 1 \pmod p$。

证明：当 i 取遍 $1 \sim p-1$ 时，$a \cdot i$ 也取遍 $1 \sim p-1$。所以

$$(p-1)! \equiv \prod_{i=1}^{p-1} i \equiv \prod_{i=1}^{p-1} ai \equiv a^{p-1}(p-1)! \pmod p,$$

因此 $a^{p-1} \equiv 1 \pmod p$。

推论：$a^n \equiv a^{n \bmod (p-1)} \pmod p$。

快速幂求逆元：$a^{p-1} \equiv a^{p-2}a \equiv 1 \pmod p$，所以 $a^{-1} \equiv a^{p-2} \pmod p$。

最短最好写，但是只能求质数的（如果不是专门的数论题一般够用了）。

Code (逆元 3)

```
// a^x (mod p), pw(a) 表示 a 的逆元
ll pw(ll a, ll x = P - 2) {
    ll ret = 1;
    for (; x; x >>= 1, a = a * a % P)
        if (x & 1) ret = ret * a % P;
    return ret;
}
```

暴力判质数的复杂度是 $O(\sqrt{n})$。

费马小定理：p 是质数 $\Rightarrow a^{p-1} \bmod p = 1$。 $a^{p-1} \bmod p = 1 \Rightarrow p$ 是质数？

强伪素数：对所有互质的 a 都满足 $a^{m-1} \bmod m = 1$ 的合数 m，比如 $m = 561 = 3 \times 11 \times 17$。

对于质数 p，若 $x^2 \equiv 1 \pmod p$，则 $x \equiv \pm 1 \pmod p$。

Miller-Rabin 素性测试：求 $a^{m-1}, a^{(m-1)/2}, \dots, a^{(m-1)/2^k}$，直到 $a^{(m-1)/2^k} \bmod p \neq 1$，判断这个结果是不是 -1，如果是 -1 就认为是质数。

只需要选前 9 个质数作为 a 就可以测试 10^{18} 以内的数。

复杂度 $O(\log n)$。

简化剩余系

模 m 下的简化剩余系：取遍所有与 m 互质的数，即 $Z_m = \{x : 1 \le x < m,\ \gcd(x, m) = 1\}$。

对于任意满足 $\gcd(a, m) = 1$ 的 a（即 $a \in Z_m$），若 i 取遍 Z_m，则 $i \cdot a$ 取遍 Z_m。

欧拉函数

欧拉函数：简化剩余系的大小，即 $\phi(m) = |Z_m|$。

性质：

- $\phi(p) = p - 1$（p 与除倍数外的所有数互质）；
- 若 $\gcd(a, b) = 1$，则 $\phi(ab) = \phi(a)\phi(b)$（即 ϕ 是积性函数）；
- 对于质数 p，$\phi(p^e) = (p-1)\,p^{e-1}$；
- $\phi(n) = n\prod_{p|n}(1 - 1/p)$，其中 p 是质数。

欧拉定理

对于任意互质的 a, m，$a^{\phi(m)} \equiv 1 \pmod{m}$。

证明：由于 $\gcd(a, m) = 1$，当 i 取遍 Z_m 时，$a \cdot i$ 也取遍 Z_m。所以

$$\prod_{i \in Z_m} i \equiv \prod_{i \in Z_m} ai \equiv a^{\phi(m)} \prod_{i \in Z_m} i \pmod{m},$$

因此 $a^{\phi(m)} \equiv 1 \pmod{m}$。

推论：$a^n \equiv a^{n \bmod \phi(m)} \pmod{p}$。

拓展欧拉定理

拓展欧拉定理

对于任意 a, n, m，当 $n \geq \phi(m)$ 时，$a^n \equiv a^{(n \bmod \phi(m))+\phi(m)} \pmod{m}$。

即 $a^n \equiv a^{\min\{n,\ (n \bmod \phi(m))+\phi(m)\}} \pmod{m}$。

可以对任意模数把次数从 n 降到 $2\phi(m)$ 以内。

例题 2

Luogu P4139　上帝与集合的正确用法

给定 m，序列 a 满足 $a_0 = 2$，$a_{i+1} = 2^{a_i}$。求 a_i 收敛到哪个值。

范围：10^3 组询问，$p \le 10^7$

令 $A = 2^{2^{2^{\cdots}}}$，则 $A \bmod m = 2^{(A \bmod \phi(m)) + \phi(m)} \bmod m$。

于是问题就转化为求 $A \bmod \phi(m)$，递归求解即可。

例题 3

SDOI2010　古代猪文

求 $g^{\sum_{k|n}\binom{n}{k}} \bmod p$，其中 $p = 999911659$ 是一个质数。

范围：$n, g \le 10^9$。

这题要用到 Lucas 定理：$\binom{n}{m} = \binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor}\binom{n \bmod p}{m \bmod p} \pmod p$。定理成立条件是 p 是质数。

用欧拉定理可以转化为指数对 $\phi(p) = 2 \times 3 \times 4679 \times 35617$ 取模，但是 $\phi(p)$ 并不是质数。

注意到 $2, 3, 4679, 35617$ 都是质数，因此可以分别计算 $\sum_{k|n}\binom{n}{k}$ 对这些质数取模的结果，最后用 CRT 合并。

SDOI2010　古代猪文

求 $g^{\sum_{k|n}\binom{n}{k}} \bmod p$，其中 $p = 999911659$ 是一个质数。

范围：$n, g \le 10^9$。

这题要用到 Lucas 定理：$\binom{n}{m} = \binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor}\binom{n \bmod p}{m \bmod p} \pmod p$。定理成立条件是 p 是质数。

用欧拉定理可以转化为指数对 $\phi(p) = 2 \times 3 \times 4679 \times 35617$ 取模，但是 $\phi(p)$ 并不是质数。

注意到 $2, 3, 4679, 35617$ 都是质数，因此可以分别计算 $\sum_{k|n}\binom{n}{k}$ 对这些质数取模的结果，最后用 CRT 合并。

一些模合数的题目可以对模数分解质因数，单独处理每个质数幂，最后 CRT 合并。

积性函数

- 积性函数：若 $f(1)=1$ 且对任意互质的 a,b 满足 $f(ab)=f(a)f(b)$，则称 f 是积性函数。
- 完全积性函数：若 $f(1)=1$ 且对任意 a,b 满足 $f(ab)=f(a)f(b)$，则称 f 是积性函数。

比如欧拉函数、因子个数、除数函数（$\sigma(n)=\sum_{d|n} d$）、恒等函数（完全积性函数）。

埃拉托斯特尼筛（埃氏筛）：给每个没被打标记的数的所有倍数（不包括自己）打标记，那么最后所有没有标记的数就是质数。

Code

```
// 求 n 以内的所有质数
vector<int> Eratosthenes(int n) {
    vector<int> vis(n + 1);
    vector<int> prime;
    for (int i = 2; i <= n; ++i) {
        if (vis[i]) continue;
        prime.push_back(i);    // i 是质数
        for (int j = i * 2; j <= n; j += i)
            vis[j] = 1;        // i 的倍数都不是素数
    }
    return prime;
}
```

复杂度：$\sum_{i:p_i \le n} n/p_i = O(n \log \log n)$。

如果只需要筛质数一般用埃氏筛就够了。

埃式筛中有很多无效操作，比如 6 会被 2 筛一遍，之后又会被 3 筛一遍。

欧拉筛（线性筛）：保证每个合数只会被**最小质因数**筛掉。复杂度 $O(n)$。

Code

```
vector<int> Euler(int n) {
    vector<int> vis(n + 1);
    vector<int> prime;
    for (int i = 2; i <= n; ++i) {
        if (!vis[i]) {
            // i 是质数
            prime.push_back(i);
        }
        for (int p : prime) {
            if (i * p > n) break;
            // 枚举不超过 i 的所有质数
            vis[i * p] = 1;
            if (i % p == 0) {
                // 之后的 p 都不满足 p ≤ i 的最小质因子
                break;
            }
        }
    }
    return prime;
}
```

欧拉筛

欧拉筛可以用来求积性函数。

假设要求积性函数 f 的值，且 $f(p)$ 和 $f(p^e)$ 已知。

访问 pi 时，p 一定不超过 i 的最小质因子，所以可以分两类讨论：

- $p \nmid i$：此时 p 与 i 互质，所以 $f(pi) = f(p)f(i)$。
- $p \mid i$：假设 $i = p^e i'$ $(\gcd(p^e, i') = 1)$，则 $f(pi) = f(p^{e+1})f(i/p^e)$。

Code (线性筛求欧拉函数)

```
vector<int> Euler(int n) {
    vector<int> vis(n + 1);
    vector<int> prime;
    vector<int> f(n + 1);
    vector<int> cnt(n + 1);  // 记录最小质因子数量 (这里没用)
    f[1] = 1;
    for (int i = 2; i <= n; ++i) {
        if (!vis[i]) {
            prime.push_back(i);
            f[i] = i - 1;  // f(质数) 的值
            cnt[i] = 1;    // 质数只有一个因子
        }
        for (int p : prime) {
            if (i * p > n) break;
            vis[i * p] = 1;
            if (i % p != 0) {
                cnt[i * p] = 1;
                f[i * p] = f[i] * f[p];   // 此时 p < i 的最小质因子, 所以 gcd(i,p) = 1
            } else {
                cnt[i * p] = 1 + cnt[i];  // i 增加一个最小质因子
                f[i * p] = f[i] * p;      // ϕ(p^e) = p · ϕ(p^(e−1))
                break;
            }
        }
    }
    return f;
}
```

例题 1

Luogu P1835　素数密度

求 $[\ell, r]$ 内的质数个数。

范围：$r \le 10^{10}, r - \ell \le 10^6$

n 的最小质因子不超过 $\sqrt{n}$，所以只要求出 10^5 以内的质数，就可以把 10^{10} 以内的所有质数都筛出来。

但我们显然不需要筛完所有数。注意到 $[\ell, r]$ 的长度最多只有 10^6，所以可以只在这个区间内筛。复杂度

$$\sum_{i:p_i \le \sqrt{r}} \frac{r-\ell}{p_i} = O((r-\ell)\log\log r)$$

例题 2

Luogu P2568　GCD

求有多少 x, y 满足 $1 \le x, y \le n$ 且 $\gcd(x, y)$ 是质数。

范围：$n \le 10^7$

固定质数 p，则我们需要使 $\gcd(x, y) = p$，即 $\gcd(x/p, y/p) = 1$。

令 $n' = \lfloor n/p \rfloor$，则满足条件的 (x, y) 数量为

$$\sum_{i=1}^{n'} \sum_{j=1}^{n'} [\gcd(i, j) = 1] = 2 \sum_{i=1}^{n'} \sum_{j=1}^{i} [\gcd(i, j) = 1] - 1 = 2 \sum_{i=1}^{n'} \phi(i) - 1.$$

线性筛预处理 ϕ 的前缀和，最后枚举所有质数加起来即可。复杂度 $O(n)$。

Thanks