

# 简单数论

July 23, 2019

- 对于两个整数  $a, b$ , 存在两个唯一的整数  $q, r$  满足:

$$b = aq + r (0 \leq r < |a|)$$

- 当  $r = 0$  时, 我们称  $a$  整除  $b$ , 记作  $a|b$ 。

- 对于两个正整数  $a, b$ , 如果有  $a|b$ , 那么称  $a$  是  $b$  的约数。
- 称一个数为质数当且仅当这个数的质因子只有 1 和他本身。  
特别地, 1 不是质数。

# 算术基本定理

- $n$  的分解唯一。
- 正确性显然。证明不显然。

- 素数无限。
- $\lim_{n \rightarrow \infty} \frac{\pi(n) * \ln n}{n} = 1$
- $P_n = O(n \log n)$
- $\sum_{i=1}^n \frac{1}{i} = O(\log n)$
- $\sum_{1 \leq p \leq n} \frac{1}{p} = O(\log \log n)$

- 假设大家都会一些简单的整除的性质
- $a|c, b|c, (a, b) = 1 \Rightarrow ab|c$
- $a|bc, (a, b) = 1 \Rightarrow a|c$
- $p|ab \Rightarrow p|a$  或  $p|b$

- 对于  $a, b$ , 如果  $d|a, d|b$ , 称  $d$  是  $a, b$  的公约数。
- 对于其中最大的  $d$ , 我们称  $d$  为  $a, b$  的最大公约数。记为  $d = (a, b)$ 。
- 同样地, 我们对于  $a, b$ , 如果  $a|d, b|d$ , 称  $d$  是  $a, b$  的公倍数。
- 对于其中最小的  $d$ , 我们称  $d$  为  $a, b$  的最小公约数。记为  $d = [a, b]$ 。
- 本质就是质因数分解后他们指数的关系。

- 算最大公约数。
- $(a, b) = (a - b, b) = (a \bmod b, b)$
- 每次取模会有一个数减小一半，复杂度是  $O(\log(a + b))$ 。



- 任意整数  $a, b, d, (a, b) \mid d \iff \exists$  整数  $u, v$  使得  $ua + vb = d$

# 扩展欧几里得算法

- $a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$
- 由归纳假设存在  $u', v'$  使得  $u'b + v'(a \bmod b) = d$ 。
- 即  $u'b + v'(a - \lfloor \frac{a}{b} \rfloor b) = d$
- $v'a + (u' - \lfloor \frac{a}{b} \rfloor v')b = d$
- 于是就得到了  $(a, b)$  的解。

- 小凯手中有两种面值的金币，两种面值均为正整数且彼此互素。每种金币小凯都有无数个。在不找零的情况下，仅凭这两种金币，有些物品他是无法准确支付的。现在小凯想知道在无法准确支付的物品中，最贵的价值是多少金币？
- 设两个面额为  $a, b$ ，那么有  $(a, b) = 1$ 。其实就是找一个最大的  $d$ ，使得  $ua + vb = d$  没有非负整数解。
- 取  $u = (b - 1), v = -1$  或  $u = -1, v = (a - 1)$  即可。

- $a \equiv b \pmod{m} \iff m \mid b - a$
- $a \equiv b \pmod{m}, a \equiv b \pmod{n} \Rightarrow a \equiv b \pmod{[m, n]}$
- $(k, m) = d, ka \equiv ka' \pmod{m} \Rightarrow a \equiv a' \pmod{\frac{m}{d}}$

# 解线性同余方程

- $ax \equiv b \pmod{m}, ax + my = b$  两者等价。
- 一开始我们有  $mx \equiv 0 \pmod{m}, ax \equiv b \pmod{m}$ , 然后对  $x$  前面的系数辗转相除即可。

- 如果有  $(b, m) = 1$ , 那么存在  $b^{-1}$  使得  $bb^{-1} \equiv 1 \pmod{m}$ 。
- 证明: 裴蜀定理
- 如果求  $\frac{a}{b} \pmod{m}$ , 可以转化成  $a * b^{-1} \pmod{m}$

- 任何  $m$  个分别属于  $m$  个剩余类的数组成剩余系。

- 所有的  $n$  满足  $0 < n \leq m, (n, m) = 1$  构成了一个模  $m$  的简化剩余系。
- 记这样  $n$  的个数为  $\varphi(m)$ 。



- 如果  $(m, m') = 1$ ,  $a$  取遍模  $m$  的简化剩余系,  $a'$  取遍模  $m'$  的简化剩余系, 那么  $am' + a'm$  取遍模  $mm'$  的简化剩余系。
- 证明略。
- 如果  $(m, m') = 1$ , 那么  $\varphi(mm') = \varphi(m)\varphi(m')$
- $\varphi(p^e) = (p-1) * p^{e-1}$
- $\varphi(m) = m \prod_{p|m} (1 - \frac{1}{p})$

- 如果  $(a, m) = 1$ , 那么  $a^{\varphi(m)} \equiv 1 \pmod{m}$
- 证明: 当  $x$  取遍模  $m$  的简化剩余系时,  $ax$  也取遍模  $m$  的简化剩余系。
- $\prod x \equiv \prod(ax) \pmod{m}$
- $\prod(ax) = a^{\varphi(m)} \prod x$

- 由于  $a^{\phi(m)} \equiv 1 \pmod{m}$ , 那么  $a^{-1} \equiv a^{\phi(m)-1} \pmod{m}$
- 如果  $m$  为素数, 那么答案为  $a^{m-2}$
- 否则需将  $m$  分解, 或解线性同余方程。

# 求 $1 \sim n$ 逆元

- 记  $f_i = i! \bmod p, g_i = (i!)^{-1} \bmod p$
- 容易发现  $g_i = g_{i+1} * (i+1), i^{-1} = f_{i-1} * g_i$
- 只需要算出  $f_n$ , 然后求出  $f_n$  的逆元  $g_n$ , 然后递推即可。
- 这个方法可以推广到  $a_1, a_2, \dots, a_n$  的逆元上。
- $n \bmod i = n - \lfloor \frac{n}{i} \rfloor i$
- 两边取逆有  $i^{-1} \equiv -\lfloor \frac{n}{i} \rfloor (n \bmod i)^{-1}$

# 线性同余方程组

- $x \equiv a_i \pmod{m_i}$
- 中国剩余定理, Chinese Remainder Theorem, 简称 CRT。
- 如果  $m_i$  两两互质,  $x \equiv \sum_{i=1}^k M_i * N_i * a_i$ , 其中  $m = \prod_{i=1}^k m_i$ ,  $M_i = \frac{m}{m_i}$ ,  $M_i * N_i \equiv 1 \pmod{m_i}$
- 其中的思想, 每一部分只对自己的方程有影响, 不改变其他方程的答案。

# 线性同余方程组

- 增量法
- 一开始有两个方程  $x \equiv a \pmod{b}, x \equiv c \pmod{d}$ 。
- 那么有  $bt + a \equiv c \pmod{d}$ ，转化成线性同余方程。
- 然后解出  $t \equiv t_0 \pmod{\frac{d}{(d,b)}}$
- 得出  $x \equiv x_0 \pmod{[b, d]}$
- 然后加入下一条方程。
- 可以解决模数不互质的情况。
- 两种方法各有千秋。

- 一般对于合数等一些不好处理的情况时，可以考虑分解成若干个素数的幂，然后用解线性同余方程组合并。

# 欧拉定理的推广

- 许多求  $a^b \bmod m$  可以转化成  $b \bmod \varphi(m)$ 。
- 对于一个  $(a, m) \neq 1$  情况的一个推广
- $a^b \equiv a^{\min(b \bmod \varphi(m) + \varphi(m), b)} \pmod{m}$



- 求  $2^{2^{2^{\dots}}} \bmod p$ 。
- 做法就是反复使用推广的欧拉定理。

- 给  $a_1, a_2, \dots, a_n$ , 求  $a_1^{a_2^{a_3^{\dots}}}$  mod  $p$ 。
- 做法依旧是反复套用之前的欧拉定理, 但需要特殊处理数值小的情况。

# $a \pmod m$ 的阶

- 如果  $(a, m) = 1$ , 那么记  $x$  为最小的正整数使得  $a^x \equiv 1 \pmod m$ 。
- $x \mid \varphi(m)$ , 反证法。
- 所以我们求阶的时候可以从  $\varphi(m)$  开始依次除掉每个因子判断。

# $n!$ 中 $p$ 的指数

- $\sum_{i \geq 1} \lfloor \frac{n}{p^i} \rfloor$
- $\frac{n-f(n)}{p-1}, f(n)$  表示  $n$  在  $p$  进制下的数位和。

- $\binom{a}{b} \bmod m$
- 对于  $m$  为大素数，如果  $a, b$  很小可以考虑杨辉三角直接递推。
- 对于  $m$  为大素数，如果  $a, b$  稍大可以暴力  $f_a * g_b * g_{a-b}$  即可。
- 一般做题中遇到的都是这两种情况。

- 对于  $m$  为小素数, Lucas 定理。
- 对于  $m$  为小素数幂, 分治。

- 求  $G^{\sum_{k|n} \binom{n}{k}} \bmod p$ 。
- $p = 999911659$
- 通过欧拉定理转化成指数对  $\varphi(p) = 2 * 3 * 4679 * 35617$  取模的值。
- 然后分解因数，对于每个因子使用 Lucas 定理，使用中国剩余定理合并即可。

- 求  $\frac{(a_1+a_2+\dots+a_k)!}{a_1!a_2!\dots a_k!} \bmod m$ 。
- 满足  $m = \prod_{i=1}^n p_i^{e_i}, p_i^{e_i} \leq 10^5$



# long long 相乘对 long long 取模

- 快 (man) 速乘
- $(x*y - (ll)((long double)x*y)/mod)*mod \% mod$

# 积性函数

- 对于  $(a, b) = 1, f(ab) = f(a)f(b)$ , 那么  $f(x)$  为积性函数。
- 前面已经证明  $\varphi(x)$  为积性函数。
- 常见的积性函数有  $d(x), \sigma(x), id(x), e(x), I(x), \mu(x)$
- $d(x) = \sum_{a|x} 1, \sigma(x) = \sum_{a|x} a, id(x) = x, I(x) = 1$
- $e(x) = 1 \iff x = 1$

- 用来求  $1 \sim n$  里面的素数或者积性函数的值。
- 用埃氏筛法筛素数的时间复杂度是  $O(n \log \log n)$ 。
- 欧式筛法的时间复杂度是  $O(n)$  的。
- 同时可以筛一些  $\varphi$  与  $\mu$ 。

- 令  $f(i)$  表示将  $i$  表示成  $a \times b \times c$  的方案数。
- 求  $f(1), \dots, f(n)$ 。

- 有  $a^{p-1} \equiv 1 \pmod{p}$ ，所以我们想到随机一个数字  $a$ ，然后判断  $a^{m-1} \pmod{m}$  是否为 1。
- 可惜我们会碰到强伪素数  $m$ ，满足费马小定理，即  $a^{m-1} \equiv 1 \pmod{m}$  对所有  $a$  都成立。
- $m = 561$  就是强伪素数。
- 对于素数  $p$ ，我们有如果  $x^2 \equiv 1 \pmod{p}$ ，那么有  $x \equiv \pm 1 \pmod{p}$ 。
- 我们可以随机几个数字  $a$ ，计算  $a^{(p-1)/2^m}$ ，然后不断平方，看它变成 1 的时候前一项是否为  $-1$ 。
- 对于  $10^{18}$  级别的数字，只需要选前九个素数即可。

- 见黑板吧。

- 对于一个度数为  $n$  的多项式  $F(x)$ , 那么  $F(x) \equiv 0 \pmod{p}$  至多  $\min(n, p)$  个解。

- $(p-1)! \equiv -1 \pmod{p}$
- 考虑  $2, 3, \dots, p-2$  这些数都存在逆元，可以两两匹配。



# 扩展扩展欧几里得算法

- 求  $\sum_{i=0}^n \lfloor \frac{ai+b}{c} \rfloor$
- 不妨设  $a, b < c$ , 所以有
- 

$$\begin{aligned}\sum_{i=0}^n \lfloor \frac{ai+b}{c} \rfloor &= \sum_{i=0}^n \sum_{0 \leq j < \lfloor \frac{ai+b}{c} \rfloor} 1 \\&= \sum_{j=0}^{\lfloor \frac{an+b}{c} \rfloor - 1} \sum_{i=0}^n [j < \lfloor \frac{ai+b}{c} \rfloor] \\&= \sum_{j=0}^{\lfloor \frac{an+b}{c} \rfloor - 1} \sum_{i=0}^n [i > \lfloor \frac{cj+c-b-1}{a} \rfloor] \\&= \sum_{j=0}^{\lfloor \frac{an+b}{c} \rfloor - 1} n - \lfloor \frac{cj+c-b-1}{a} \rfloor\end{aligned}$$

- 求  $\bigoplus_{i=0}^n (a + di)$ , 其中  $a + dn < 2^{32}$ 。
- 对每一位求奇偶性。

- 如果  $g \pmod{m}$  的阶为  $\varphi(m)$ , 那么  $g$  为  $m$  的原根。
- $g^0, g^1, \dots, g^{\varphi(m)-1}$  构成了模  $m$  的简化剩余系。
- 只有  $1, 2, 4, p^a, 2p^a$  存在原根。

- 从小到大枚举或随机  $g$ ，然后判断是否为原根。
- 对于素数  $a$ ，只要判断所有的  $p|a-1$ ，然后  $g^{\frac{a-1}{p}} \not\equiv 1 \pmod{a}$  即可。
- 如果  $g$  是  $p$  的原根，那么它一定是  $p^e$  的原根。

- $a^x \equiv b \pmod{m}$
- 如果  $m$  为素数, 用 baby-step giant-step 算法。
- 可以知道  $x \equiv x_0 \pmod{\varphi(m)}$
- 令  $x = q * t - r$ , 预处理  $a^{t*q}$ 。
- 查询时枚举  $r$ , 然后看是否存在  $b * a^r$ , 用 hash 表或者 map 检索。
- 预处理复杂度  $O(\frac{m}{t})$ , 查询复杂度  $O(t)$ 。
- 如果  $m$  不是素数, 提取公因子, 使得  $a, m$  互质, 然后用 BSGS 解决。

# 指数方程

- $x^a \equiv b \pmod{m}$
- 如果  $m$  为素数，先求出  $m$  的原根  $g$ 。
- 解出  $g^s \equiv b \pmod{m}$
- 令  $x = g^t$ ，那么原方程化为  $g^{ta} \equiv g^s \pmod{m}$ 。
- 由于原根的性质，等价于  $ta \equiv s \pmod{\varphi(m)}$
- 如果  $m$  不为素数，先分解，然后用 CRT 合并。
- 注意  $2^n$  没有原根，可以枚举。

# 指数方程

- $x^a \equiv b \pmod{m}$
- 如果  $m$  为素数, 且  $(a, \varphi(m)) = 1$
- 那么求出  $a$  模  $\varphi(m)$  的逆元  $a^{-1}$
- $x^{aa^{-1}} \equiv x \equiv b^{a^{-1}} \pmod{m}$

# 积性函数

- 对于  $(a, b) = 1, f(ab) = f(a)f(b)$ , 那么  $f(x)$  为积性函数。
- 前面已经证明  $\phi(x)$  为积性函数。
- 常见的积性函数有  $d(x), \sigma(x), id(x), e(x), I(x), \mu(x)$
- $d(x) = \sum_{a|x} 1, \sigma(x) = \sum_{a|x} a, id(x) = x, I(x) = 1$
- $e(x) = 1 \iff x = 1$



- 两个数论函数  $f(x), g(x)$ , 令  $h = f * g$
- 那么  $h(x) = \sum_{a|x} f(a)g(\frac{x}{a})$
- 容易证明卷积满足交换律, 结合律。
- 两个积性函数的卷积还是积性函数。
- $f * e = f$
- 注意卷积不一定要求两个函数都是积性函数。

# 莫比乌斯函数

- $\mu(n) = (-1)^k \iff n = \prod_{i=1}^k p_i$
- 如果  $n$  有平方因子, 那么  $\mu(n) = 0$
- $\mu * I = e$ , 即  $\sum_{d|n} \mu(d) = e(n)$
- 由于两个积性函数的卷积还是积性函数, 所以考虑每个素数幂即可。

- 如果  $F(n) = \sum_{d|n} f(d)$ , 那么  $f(n) = \sum_{d|n} \mu(d) * F(\frac{n}{d})$
- $F = f * I \Rightarrow F * \mu = (f * I) * \mu = f * (I * \mu) = f * e = f$
- 这里不要求  $f$  为积性函数。

$$\sum_{i=1}^n d(i)$$

- $d = I * I$ , 即  $d(n) = \sum_{xy=n} 1$
- $\sum_{i=1}^n d(i) = \sum_{xy \leq n} 1 = \sum_{x=1}^n \lfloor \frac{n}{x} \rfloor$
- $\lfloor \frac{n}{x} \rfloor$  只有  $O(\sqrt{n})$  段, 这将成为很多暴力算法时间复杂度的基础。

$$\sum_{i=1}^n \sum_{j=1}^m (i, j)$$

- 不妨设  $n \leq m$
- 枚举最大公约数  $d$ , 求有多少对  $(i, j)$  满足  $(i, j) = d$
- $\sum_{d=1}^n d \sum_{i \leq n, j \leq m} [(i, j) = d]$
- $\sum_{d=1}^n d \sum_{i \leq n, j \leq m} [(\frac{i}{d}, \frac{j}{d}) = 1]$
- $\sum_{d=1}^n d \sum_{i \leq n, j \leq m} e((\frac{i}{d}, \frac{j}{d}))$
- $\sum_{d=1}^n d \sum_{i \leq n, j \leq m} \sum_{d' | \frac{i}{d}, d' | \frac{j}{d}} \mu(d')$
- $\sum_{dd'} d * \mu(d') \sum_{dd' | i, dd' | j} 1$
- 令  $t = dd'$
- $\sum_{t=1}^n \lfloor \frac{n}{t} \rfloor \lfloor \frac{m}{t} \rfloor (id * \mu)(t)$
- $\sum_{t=1}^n \lfloor \frac{n}{t} \rfloor \lfloor \frac{m}{t} \rfloor \phi(t)$

- $\sum_{t=1}^n \lfloor \frac{a}{t} \rfloor \lfloor \frac{b}{t} \rfloor \lfloor \frac{c}{t} \rfloor \sum_{d|t} q^d \mu(\frac{t}{d})$

- $\sum_{1 \leq i \leq n, 1 \leq j \leq m, i \neq j} (n \% i)(m \% j)$
- 不妨设  $i \leq j$
- $\sum_{i=1}^n (n \% i) \sum_{j=1}^m (m \% j) - \sum_{i=1}^n (n \% i)(m \% i)$
- $\sum_{i=1}^n (n \% i) = \sum_{i=1}^n n - \lfloor \frac{n}{i} \rfloor * i$
- $\sum_{i=1}^n (n \% i)(m \% i) = \sum_{i=1}^n (n - \lfloor \frac{n}{i} \rfloor * i)(m - \lfloor \frac{m}{i} \rfloor * i)$
- $\sum_{i=1}^n nm - (n \lfloor \frac{m}{i} \rfloor + m \lfloor \frac{n}{i} \rfloor) * i + \lfloor \frac{n}{i} \rfloor \lfloor \frac{m}{i} \rfloor * i^2$

- Maintain an array  $a$  with index from 1 to  $l$ . There are two kinds of operations:
- 1. Add  $v$  to  $a_x$  for every  $x$  that  $(x, n) = d$ .
- 2. Query  $\sum_{i=1}^x a_x$



- 建立一个辅助数组  $f$ , 使得  $a_i = \sum_{d|i} f_d$ 。
- $[gcd(x, n) = d] * v = [gcd(\frac{x}{d}, \frac{n}{d}) = 1] * v = \sum_{p|\frac{x}{d}, p|\frac{n}{d}} \mu(p) * v$
- 所以对所有  $p|\frac{n}{d}$ , 我们只要对  $f_{pd}$  加上  $\mu(p) * v$  就行了。
- 对于操作 2,  $\sum_{i=1}^x a_i = \sum_{d=1}^x [\frac{x}{d}] f(d)$
- 对于所有  $d < \sqrt{x \log x}$ , 暴力计算,  $d \geq \sqrt{x \log x}$ ,  $[\frac{x}{d}]$  至多有  $\sqrt{\frac{x}{\log x}}$  段。所以用树状数组维护  $f$  的前缀和, 分段统计就行了。
- 时间复杂度  $O(\sqrt{x \log x})$ 。