

# 类欧几里得算法

洪华敦

WC2016



- $[a]$  若表达式 $a$ 为真，则 $[a] = 1$ ，否则 $[a] = 0$

- $[a]$  若表达式 $a$ 为真, 则 $[a] = 1$ , 否则 $[a] = 0$
- 例如 $[1 + 1 = 2] = 1$

- $[a]$  若表达式 $a$ 为真, 则 $[a] = 1$ , 否则 $[a] = 0$
- 例如 $[1 + 1 = 2] = 1$
- $\%$  取模

- $[a]$  若表达式 $a$ 为真, 则 $[a] = 1$ , 否则 $[a] = 0$
- 例如 $[1 + 1 = 2] = 1$
- $\%$  取模
- $S_d(n) = \sum_{x=1}^n x^d$

- $[a]$  若表达式 $a$ 为真, 则 $[a] = 1$ , 否则 $[a] = 0$
- 例如 $[1 + 1 = 2] = 1$
- $\%$  取模
- $S_d(n) = \sum_{x=1}^n x^d$
- $\lfloor \frac{a}{b} \rfloor$   $\frac{a}{b}$  下取整





- 对于  $b \neq 0$ , 有  $\gcd(a, b) = \gcd(b, a \% b)$



- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度

- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度
- 考虑最坏情况， $a \leq 2 * b - 1$

- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度
- 考虑最坏情况， $a \leq 2 * b - 1$
- 那么有 $\gcd(a, b) = \gcd(b, a - b)$

- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度
- 考虑最坏情况， $a \leq 2 * b - 1$
- 那么有 $\gcd(a, b) = \gcd(b, a - b)$
- 将 $a, b$ 设为比较接近的斐波那契数，那么 $a = F_n$ ， $b = F_{n-1}$ ，则有 $\gcd(F_n, F_{n-1}) = \gcd(F_{n-1}, F_{n-2})$

- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度
- 考虑最坏情况， $a \leq 2 * b - 1$
- 那么有 $\gcd(a, b) = \gcd(b, a - b)$
- 将 $a, b$ 设为比较接近的斐波那契数，那么 $a = F_n$ ， $b = F_{n-1}$ ，则有 $\gcd(F_n, F_{n-1}) = \gcd(F_{n-1}, F_{n-2})$
- 显然 $n$ 是 $O(\log(F_n))$ 的

- 为了更好地讲后面的东西，我们来证明下欧几里得算法的复杂度
- 考虑最坏情况， $a \leq 2 * b - 1$
- 那么有 $\gcd(a, b) = \gcd(b, a - b)$
- 将 $a, b$ 设为比较接近的斐波那契数，那么 $a = F_n$ ， $b = F_{n-1}$ ，则有 $\gcd(F_n, F_{n-1}) = \gcd(F_{n-1}, F_{n-2})$
- 显然 $n$ 是 $O(\log(F_n))$ 的
- 所以这个算法的复杂度是 $O(\log(a))$ 的



类欧几里得算  
法

洪华敦

需要用的的一  
些数学符号

关于欧几里得  
算法

类欧几里得算  
法

一些简单的扩  
展

- 我们来看这样一个问题

- 我们来看这样一个问题
- $\sum_{x=0}^n \left\lfloor \frac{ax+b}{c} \right\rfloor$



- 首先, 如果  $a \geq c$ , 设  $d = \lfloor \frac{a}{c} \rfloor$  则有:

- 首先, 如果  $a \geq c$ , 设  $d = \lfloor \frac{a}{c} \rfloor$  则有:
- $$\sum_{x=0}^n \lfloor \frac{ax+b}{c} \rfloor = \sum_{x=0}^n \lfloor \frac{(a\%c)x+b}{c} \rfloor + d * x$$

- 首先, 如果  $a \geq c$ , 设  $d = \lfloor \frac{a}{c} \rfloor$  则有:
- $\sum_{x=0}^n \lfloor \frac{ax+b}{c} \rfloor = \sum_{x=0}^n \lfloor \frac{(a\%c)x+b}{c} \rfloor + d * x$
- $= d * S_1(n) + \sum_{x=0}^n \lfloor \frac{(a\%c)x+b}{c} \rfloor$





- 同理，如果有  $b \geq c$ ，设  $d = \lfloor \frac{b}{c} \rfloor$  则有：

- 同理，如果有  $b \geq c$ ，设  $d = \lfloor \frac{b}{c} \rfloor$  则有：
- $$\sum_{x=0}^n \lfloor \frac{ax+b}{c} \rfloor = \sum_{x=0}^n \left\lfloor \frac{ax+(b\%c)}{c} \right\rfloor + d$$

- 同理，如果有  $b \geq c$ ，设  $d = \lfloor \frac{b}{c} \rfloor$  则有：
- $\sum_{x=0}^n \lfloor \frac{ax+b}{c} \rfloor = \sum_{x=0}^n \left\lfloor \frac{ax+(b\%c)}{c} \right\rfloor + d$
- $= d * S_0(n) + \sum_{x=0}^n \left\lfloor \frac{ax+(b\%c)}{c} \right\rfloor$

类欧几里得算  
法

洪华敦

需要用的的一  
些数学符号

关于欧几里得  
算法

类欧几里得算  
法

一些简单的扩  
展

- 经过上面两个计算后，我们有  $a < c$ ,  $b < c$

- 经过上面两个计算后, 我们有  $a < c$ ,  $b < c$
- 于是我们的问题规模从  $(a, c, b, n)$  变成了  $(a\%c, c, b\%c, n)$



- 令  $M = \lfloor \frac{an+b}{c} \rfloor$



- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$

- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$
- 我们来化简下这个式子  $y < \lfloor \frac{ax+b}{c} \rfloor$

- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$
- 我们来化简下这个式子  $y < \lfloor \frac{ax+b}{c} \rfloor$
- $c * (y + 1) \leq a * x + b$

- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$
- 我们来化简下这个式子  $y < \lfloor \frac{ax+b}{c} \rfloor$
- $c * (y + 1) \leq a * x + b$
- $c * y + c - b \leq a * x$

- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$
- 我们来化简下这个式子  $y < \lfloor \frac{ax+b}{c} \rfloor$
- $c * (y + 1) \leq a * x + b$
- $c * y + c - b \leq a * x$
- $c * y + c - b - 1 < a * x$

- 令  $M = \lfloor \frac{an+b}{c} \rfloor$
- 原式  $= \sum_{x=0}^n \sum_{y=0}^M [y < \lfloor \frac{ax+b}{c} \rfloor]$
- 我们来化简下这个式子  $y < \lfloor \frac{ax+b}{c} \rfloor$
- $c * (y + 1) \leq a * x + b$
- $c * y + c - b \leq a * x$
- $c * y + c - b - 1 < a * x$
- $x > \lfloor \frac{c*y+c-b-1}{a} \rfloor$



- 所以原式 =  $\sum_{y=0}^M \sum_{x=0}^n [x > \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor]$



- 所以原式 =  $\sum_{y=0}^M \sum_{x=0}^n [x > \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor]$
- $= \sum_{y=0}^M n - \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor$

- 所以原式 =  $\sum_{y=0}^M \sum_{x=0}^n [x > \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor]$
- $= \sum_{y=0}^M n - \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor$
- $= n * (M + 1) - \sum_{y=0}^M \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor$

- 所以原式 =  $\sum_{y=0}^M \sum_{x=0}^n [x > \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor]$
- $= \sum_{y=0}^M n - \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor$
- $= n * (M + 1) - \sum_{y=0}^M \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor$
- 于是我们的问题规模从  $(a\%c, c, b\%c, n)$  变成了  $(c, a\%c, c - b\%c - 1, M)$

类欧几里得算  
法

洪华敦

需要用的的一  
些数学符号

关于欧几里得  
算法

类欧几里得算  
法

一些简单的扩  
展

- 经过上面一番转化，我们的问题规模从 $(a, c, b, n)$ 变成了 $(c, a \% c, c - b \% c - 1, M)$

- 经过上面一番转化，我们的问题规模从 $(a, c, b, n)$ 变成了 $(c, a\%c, c - b\%c - 1, M)$
- 显然当 $a$ 为0时，我们可以很轻松地计算答案

- 经过上面一番转化，我们的问题规模从 $(a, c, b, n)$ 变成了 $(c, a \% c, c - b \% c - 1, M)$
- 显然当 $a$ 为0时，我们可以很轻松地计算答案
- 所以问题规模每次都从 $(a, c)$ 变成 $(c, a \% c)$

- 经过上面一番转化，我们的问题规模从 $(a, c, b, n)$ 变成了 $(c, a \% c, c - b \% c - 1, M)$
- 显然当 $a$ 为0时，我们可以很轻松地计算答案
- 所以问题规模每次都从 $(a, c)$ 变成 $(c, a \% c)$
- 这和计算 $\gcd$ 的时间复杂度是一样的



- 经过上面一番转化，我们的问题规模从 $(a, c, b, n)$ 变成了 $(c, a\%c, c - b\%c - 1, M)$
- 显然当 $a$ 为0时，我们可以很轻松地计算答案
- 所以问题规模每次都从 $(a, c)$ 变成 $(c, a\%c)$
- 这和计算 $\gcd$ 的时间复杂度是一样的
- 所以时间复杂度是 $O(\log(a))$



- 简化题意：

- 简化题意:
- 给定  $N, A, B, K, L$ , 计算:

$$\sum_{x=1}^n \binom{\lfloor \frac{Ax}{b} \rfloor}{K+1} \binom{n-x}{L}$$

- 简化题意:
- 给定  $N, A, B, K, L$ , 计算:

$$\sum_{x=1}^n \binom{\lfloor \frac{Ax}{b} \rfloor}{K+1} \binom{n-x}{L}$$

- $N, A, B \leq 10^{18}, K, L \leq 10$

- 简化题意:
- 给定  $N, A, B, K, L$ , 计算:

$$\sum_{x=1}^n \binom{\lfloor \frac{Ax}{b} \rfloor}{K+1} \binom{n-x}{L}$$

- $N, A, B \leq 10^{18}$ ,  $K, L \leq 10$
- 答案对一个大质数取模



- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式



- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式
- 设  $\left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)_{K+1} = \sum_{i=0}^{K+1} c_i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^i$

- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式
- 设  $\left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)_{K+1} = \sum_{i=0}^{K+1} c_i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^i$
- 设  $\binom{n-x}{L} = \sum_{i=0}^L d_i * x^i$

- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式
- 设  $\left\lfloor \frac{Ax}{b} \right\rfloor = \sum_{i=0}^{K+1} c_i * \left\lfloor \frac{Ax}{b} \right\rfloor^i$
- 设  $\binom{n-x}{L} = \sum_{i=0}^L d_i * x^i$
- 那么原式 =  $\sum_{x=1}^n \sum_{i=0}^L d_i * x^i \sum_{j=0}^{K+1} c_j * \left\lfloor \frac{Ax}{b} \right\rfloor^j$

- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式
- 设  $\left\lfloor \frac{Ax}{b} \right\rfloor = \sum_{i=0}^{K+1} c_i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^i$
- 设  $\binom{n-x}{L} = \sum_{i=0}^L d_i * x^i$
- 那么原式  $= \sum_{x=1}^n \sum_{i=0}^L d_i * x^i \sum_{j=0}^{K+1} c_j * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^j$
- 等于  $\sum_{i=0}^L \sum_{j=0}^{K+1} d_i * c_j \sum_{x=1}^n x^i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^j$

- 首先我们知道,  $\binom{x}{k}$  是一个关于  $x$  的  $k$  次多项式
- 设  $\left(\left\lfloor \frac{Ax}{b} \right\rfloor\right) = \sum_{i=0}^{K+1} c_i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^i$
- 设  $\binom{n-x}{L} = \sum_{i=0}^L d_i * x^i$
- 那么原式  $= \sum_{x=1}^n \sum_{i=0}^L d_i * x^i \sum_{j=0}^{K+1} c_j * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^j$
- 等于  $\sum_{i=0}^L \sum_{j=0}^{K+1} d_i * c_j \sum_{x=1}^n x^i * \left(\left\lfloor \frac{Ax}{b} \right\rfloor\right)^j$
- $d_i$  和  $c_j$  我们可以预处理用拉格朗日插值公式或者高斯消元算出来, 那么问题就变成了求后面那些东西



- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$

- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$
- 首先我们可以用二项式定理, 让  $a, b < c$



- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$
- 首先我们可以用二项式定理, 让  $a, b < c$
- 令  $M = \lfloor \frac{a*n+b}{c} \rfloor$

- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$
- 首先我们可以用二项式定理, 让  $a, b < c$
- 令  $M = \lfloor \frac{a*n+b}{c} \rfloor$
- 

$$\sum_{x=0}^n x^U * (\lfloor \frac{a * x + b}{c} \rfloor)^V$$

- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$
- 首先我们可以用二项式定理, 让  $a, b < c$
- 令  $M = \lfloor \frac{a*n+b}{c} \rfloor$

•

$$\sum_{x=0}^n x^U * (\lfloor \frac{a * x + b}{c} \rfloor)^V$$

•

$$= \sum_{x=0}^n x^U * \text{sum}_{y=0}^M [y < \lfloor \frac{a * x + b}{c} \rfloor] ((y+1)^V - y^V)$$

- 我们令  $f(a, c, b, i, j)$  表示  $\sum_{x=0}^n x^i * (\lfloor \frac{a*x+b}{c} \rfloor)^j$
- 首先我们可以用二项式定理, 让  $a, b < c$
- 令  $M = \lfloor \frac{a*n+b}{c} \rfloor$

•

$$\sum_{x=0}^n x^U * (\lfloor \frac{a * x + b}{c} \rfloor)^V$$

•

$$= \sum_{x=0}^n x^U * \text{sum}_{y=0}^M [y < \lfloor \frac{a * x + b}{c} \rfloor] ((y+1)^V - y^V)$$

•

$$= \sum_{y=0}^M ((y+1)^V - y^V) \sum_{x=0}^n x^U * [x > \lfloor \frac{c * y + c - b - 1}{a} \rfloor]$$



- 我们可以把 $(y+1)^U - y^U$ 二项式展开

- 我们可以把 $(y + 1)^U - y^U$ 二项式展开
- 原式等于

- 我们可以把 $(y+1)^U - y^U$ 二项式展开
- 原式等于
- $= \sum_{i=0}^{U-1} \binom{U}{i} \sum_{y=0}^M y^i * \sum_{x=0}^n x^U * [x > \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor]$



- 我们可以把 $(y + 1)^U - y^U$ 二项式展开
- 原式等于
- $= \sum_{i=0}^{U-1} \binom{U}{i} \sum_{y=0}^M y^i * \sum_{x=0}^n x^U * [x > \lfloor \frac{c*y+c-b-1}{a} \rfloor]$
- $= \sum_{i=0}^{U-1} \binom{U}{i} \sum_{y=0}^M y^i * (S_U(n) - S_U(\lfloor \frac{c*y+c-b-1}{a} \rfloor))$

- 我们可以把 $(y+1)^U - y^U$ 二项式展开
- 原式等于
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{x=0}^n x^U * [x > \lfloor \frac{c*y+c-b-1}{a} \rfloor]$
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * (S_U(n) - S_U(\lfloor \frac{c*y+c-b-1}{a} \rfloor))$
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * S_U(n) - \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * S_U(\lfloor \frac{c*y+c-b-1}{a} \rfloor)$

- 我们可以把 $(y + 1)^U - y^U$ 二项式展开
- 原式等于
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{x=0}^n x^U * [x > \lfloor \frac{c*y+c-b-1}{a} \rfloor]$
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * (S_U(n) - S_U(\lfloor \frac{c*y+c-b-1}{a} \rfloor))$
- $= \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * S_U(n) - \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * S_U(\lfloor \frac{c*y+c-b-1}{a} \rfloor)$
- 前者等于 $\sum_{i=0}^{V-1} \binom{V}{i} S_i(M) * S_U(n)$ , 于是问题就变成了求后者



- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d + 1$ 次多项式

- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d + 1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$

- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d + 1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组 $a$ 可以用高斯消元预处理，或者直接带伯努利数进去

- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d+1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组 $a$ 可以用高斯消元预处理，或者直接带伯努利数进去
- 原式等  
于 $\sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{j=0}^{U+1} a[U][j] * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$



- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d+1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组 $a$ 可以用高斯消元预处理，或者直接带伯努利数进去
- 原式等  
 于 $\sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{j=0}^{U+1} a[U][j] * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 等于 $\sum_{i=0}^{V-1} \sum_{j=0}^{U+1} a[U][j] * \binom{V}{i} \sum_{y=0}^M y^i * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$

- 我们可以发现  $S_d(x)$  是一个关于  $x$  的  $d+1$  次多项式
- 设  $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组  $a$  可以用高斯消元预处理，或者直接带伯努利数进去
- 原式等  
 于  $\sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{j=0}^{U+1} a[U][j] * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 等于  $\sum_{i=0}^{V-1} \sum_{j=0}^{U+1} a[U][j] * \binom{V}{i} \sum_{y=0}^M y^i * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 于是我们再次完成了  $a, c$  的互换，这样递归下去只有  $O(\log(A))$  层

- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d+1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组 $a$ 可以用高斯消元预处理，或者直接带伯努利数进去
- 原式等  

$$\text{于} \sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{j=0}^{U+1} a[U][j] * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$$
- 等于 $\sum_{i=0}^{V-1} \sum_{j=0}^{U+1} a[U][j] * \binom{V}{i} \sum_{y=0}^M y^i * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 于是我们再次完成了 $a, c$ 的互换，这样递归下去只有 $O(\log(A))$ 层
- 我们可以对于每个 $(a, b, c, n)$ 求出对于所有 $(i, j)$ 的答案，具体方法就是递归下去，然后用返回的那些答案更新

- 我们可以发现 $S_d(x)$ 是一个关于 $x$ 的 $d + 1$ 次多项式
- 设 $S_d(x) = \sum_{i=0}^{d+1} a[d][i] * x^i$
- 数组 $a$ 可以用高斯消元预处理，或者直接带伯努利数进去
- 原式等  
 于 $\sum_{i=0}^{V-1} \binom{V}{i} \sum_{y=0}^M y^i * \sum_{j=0}^{U+1} a[U][j] * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 等于 $\sum_{i=0}^{V-1} \sum_{j=0}^{U+1} a[U][j] * \binom{V}{i} \sum_{y=0}^M y^i * \left( \left\lfloor \frac{c*y+c-b-1}{a} \right\rfloor \right)^j$
- 于是我们再次完成了 $a, c$ 的互换，这样递归下去只有 $O(\log(A))$ 层
- 我们可以对于每个 $(a, b, c, n)$ 求出对于所有 $(i, j)$ 的答案，具体方法就是递归下去，然后用返回的那些答案更新
- 用一些卷积优化可以把时间复杂度优化到 $O((K + L)^3 \log(A))$