

设  $f(a,b,c,n,t_1,t_2)=\sum_{i=0}^n i^{t_1} \lfloor \frac{ai+b}{c} \rfloor^{t_2}$  求解这一函数值的算法因类似于欧几里得算法而得名类欧几里得算法。为了方便，这里仅讨论  $n$  个参数均为非负整数的情况。另外我们定义  $0^0=1$ 。现将该算法描述如下：

定义  $m = \lfloor \frac{an+b}{c} \rfloor$   $g_t(x) = (x+1)^t - x^t = \sum_{i=0}^{t-1} g_{ti} x^i$   $h_t(x) = \sum_{i=1}^x i^t = \sum_{i=0}^{t+1} h_{ti} x^i$

首先描述几种特殊情况：

- 若  $t_2=0$  那么有：

$$\begin{aligned} \text{原式} &= \sum_{i=0}^n i^{t_1} \lfloor \frac{ai+b}{c} \rfloor^0 \\ &= h_{t_1}(n) + [t_1=0] \end{aligned}$$

- 若  $m=0$  那么显然有：

$$\text{原式} = 0$$

- 若  $a=0$  那么有：

$$\begin{aligned} \text{原式} &= \sum_{i=0}^n i^{t_1} \lfloor \frac{b}{c} \rfloor^{t_2} \\ &= \lfloor \frac{b}{c} \rfloor^{t_2} (h_{t_1}(n) + [t_1=0]) \end{aligned}$$

- 若  $a \geq c$  或  $b \geq c$  那么有：

$$\begin{aligned} \text{原式} &= \sum_{i=0}^n i^{t_1} (\lfloor \frac{(a\%c)i+(b\%c)}{c} \rfloor + \lfloor \frac{a}{c} \rfloor + \lfloor \frac{b}{c} \rfloor)^{t_2} \\ &= \sum_{i=0}^n i^{t_1} \sum_{u_1+u_2+u_3=t_2} \binom{t_2}{u_1, u_2, u_3} (\lfloor \frac{(a\%c)i+(b\%c)}{c} \rfloor)^{u_1} (\lfloor \frac{a}{c} \rfloor)^{u_2} (\lfloor \frac{b}{c} \rfloor)^{u_3} \\ &= \sum_{u_1+u_2+u_3=t_2} \binom{t_2}{u_1, u_2, u_3} (\lfloor \frac{a}{c} \rfloor)^{u_2} (\lfloor \frac{b}{c} \rfloor)^{u_3} \sum_{i=0}^n i^{t_1+u_2} (\lfloor \frac{(a\%c)i+(b\%c)}{c} \rfloor)^{u_1} \\ &= \sum_{u_1+u_2+u_3=t_2} \binom{t_2}{u_1, u_2, u_3} (\lfloor \frac{a}{c} \rfloor)^{u_2} (\lfloor \frac{b}{c} \rfloor)^{u_3} f(a\%c, b\%c, c, n, t_1+u_2, u_1) \end{aligned}$$

接下来讨论一般的  $0 \leq a, b < c$  的情况

$$\begin{aligned} \text{原式} &= \sum_{i=0}^n i^{t_1} \sum_{j=1}^m g_{t_2}(j-1) \lfloor \frac{ai+b}{c} \rfloor^{t_2} \\ &= \sum_{i=0}^n i^{t_1} \sum_{j=0}^{m-1} g_{t_2}(j) \lfloor \frac{ai+b}{c} \rfloor^{t_2} \\ &= \sum_{j=0}^{m-1} g_{t_2}(j) \sum_{i=0}^n i^{t_1} \lfloor \frac{cj+c-b}{c} \rfloor^{t_2} \end{aligned}$$

显然不论如何  $i=0$  时  $\lfloor \frac{cj+c-b}{c} \rfloor \geq \lfloor \frac{c-b}{c} \rfloor > 0$  故  $\lfloor \frac{cj+c-b}{c} \rfloor$  永远为假，因此

$$\begin{aligned} \text{原式} &= \sum_{j=0}^{m-1} g_{t_2}(j) \sum_{i=1}^n i^{t_1} \lfloor \frac{cj+c-b}{c} \rfloor^{t_2} \\ &= \sum_{j=0}^{m-1} g_{t_2}(j) \sum_{i=1}^n i^{t_1} \lfloor \frac{cj+c-b-1}{c} \rfloor^{t_2} \\ &= m^{t_2} h_{t_1}(n) - \sum_{j=0}^{m-1} g_{t_2}(j) \sum_{i=1}^n i^{t_1} \lfloor \frac{cj+c-b-1}{c} \rfloor^{t_2} \end{aligned}$$

下面我们证明  $\lfloor \frac{cj+c-b-1}{c} \rfloor \leq n$

$$\lfloor \frac{cj+c-b-1}{c} \rfloor \leq \lfloor \frac{c(m-1)+c-b-1}{c} \rfloor = \lfloor \frac{cm-b-1}{c} \rfloor$$

$$\begin{aligned} & \text{假设 } \lfloor \frac{mc-b-1}{a} \rfloor > n \text{ 那么 } \\ & \begin{aligned} & mc-b-1 > na \\ & na+b+1 < mc \end{aligned} \\ & \text{而 } \lfloor \frac{na+b}{c} \rfloor = m \text{ 故 } na+b \geq mc \text{ 矛盾。故} \\ & \lfloor \frac{cj+c-b-1}{a} \rfloor \leq n \text{ 故有 } \\ & \begin{aligned} & \text{原式} = m^{\{t_2\}} h_{\{t_1\}}(n) - \\ & \sum_{j=0}^{m-1} g_{\{t_2\}}(j) \sum_{i=1}^{\lfloor \frac{cj+c-b-1}{a} \rfloor} i^{\{t_1\}} \\ & = m^{\{t_2\}} h_{\{t_1\}}(n) - \sum_{j=0}^{m-1} g_{\{t_2\}}(j) h_{\{t_1\}}(\lfloor \frac{cj+c-b-1}{a} \rfloor) \\ & = m^{\{t_2\}} h_{\{t_1\}}(n) - \sum_{j=0}^{m-1} (\sum_{u=0}^{\{t_2\}-1} g_{\{t_2\}u}^{\{u\}}) \cdot (\sum_{v=0}^{\{t_1\}+1} h_{\{t_1\}v} \lfloor \frac{cj+c-b-1}{a} \rfloor^{\{v\}}) \\ & = m^{\{t_2\}} h_{\{t_1\}}(n) - \sum_{u=0}^{\{t_2\}-1} \sum_{v=0}^{\{t_1\}+1} g_{\{t_2\}u} h_{\{t_1\}v} \sum_{j=0}^{m-1} j^{\{u\}} \lfloor \frac{cj+c-b-1}{a} \rfloor^{\{v\}} \\ & = m^{\{t_2\}} h_{\{t_1\}}(n) - \sum_{u=0}^{\{t_2\}-1} \sum_{v=0}^{\{t_1\}+1} g_{\{t_2\}u} h_{\{t_1\}v} f(c, c-b-1, a, m-1, u, v) \end{aligned} \\ & \text{易见 } (a, c) \text{ 在一次递归后变为 } (c, a \% c) \text{ 与欧几里得算法相同。故总复杂度为} \\ & \mathcal{O}(\log \max\{a, c\} (t_1 + t_2)^4) \end{aligned}$$

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

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:intrepidword:zhongzihao:similar\\_euclid&rev=1617289581](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:intrepidword:zhongzihao:similar_euclid&rev=1617289581)

Last update: 2021/04/01 23:06