# 类欧几里得

吴一岐

2020 年 5 月 21 日

# 1 描述

我们通过三个式子的化简来体会类欧几里得算法的精髓

1. $f(n,a,b,c)=\sum\limits_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor$

2. $g(n,a,b,c)=\sum\limits_{i=0}^{n}i\lfloor\frac{ai+b}{c}\rfloor$

3. $h(n,a,b,c)=\sum\limits_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor^2$

# 2 f 函数

$f(n,a,b,c)=\sum\limits_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor$

$f$ 函数的定义相当于是计算一个区域内整点的数量，即满足方程 Ax+By=C 这条直线下方的整点数量。

若 $a\geq c$ $b\geq c$

那么有

$f(n,a,b,c)=\sum\limits_{i=0}^{n}\frac{(\lfloor\frac{a}{c}\rfloor c+a\%c)i+\lfloor\frac{b}{c}\rfloor c+b\%c}{c}$

上面的式子中我们发现两个整数部分可以单独脱离出来

$$f(n,a,b,c)=\sum_{i=0}^{n}\lfloor\frac{a}{c}\rfloor i+\sum_{i=0}^{n}\lfloor\frac{b}{c}\rfloor+f(n,a\%c,b\%c,c)$$

$$f(n,a,b,c)=\frac{(n+1)n}{2}\lfloor\frac{a}{c}\rfloor+(n+1)\lfloor\frac{b}{c}\rfloor+f(n,a\%c,b\%c,c)$$

我们发现问题已经可以递归解决

若 $a<c$ $b<c$

那么令常数 $m=\lfloor\frac{an+b}{c}\rfloor$

$f(n,a,b,c)=\sum\limits_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor=\sum\limits_{i=0}^{n}\sum\limits_{j=1}^{m}[j\leq\lfloor\frac{ai+b}{c}\rfloor]$

这个式子的几何意义：一条直线与 $x=0,x=n$ 两条垂直的直线构成的一个直角梯形里面的整点的数量（x 轴上方的点）

式子的计算方法是先枚举一条竖线，再求竖线上整点的数量

我们转换成枚举横线

$f(n,a,b,c)=\sum\limits_{i=0}^{n}\sum\limits_{j=0}^{m-1}[j+1\leq\lfloor\frac{ai+b}{c}\rfloor]$

$f(n,a,b,c) = \sum_{i=0}^{n} \sum_{j=0}^{m-1} [jc + c - b \le ai]$
$f(n,a,b,c) = \sum_{i=0}^{n} \sum_{j=0}^{m-1} [jc + c - b - 1 < ai]$
$f(n,a,b,c) = \sum_{i=0}^{n} \sum_{j=0}^{m-1} [\lfloor \frac{jc+c-b-1}{a} \rfloor < i]$
$f(n,a,b,c) = \sum_{j=0}^{m-1} (n - \lfloor \frac{jc+c-b-1}{a} \rfloor)$
$f(n,a,b,c) = nm - f(m-1, c, c-b-1, a)$
之后递归处理即可

# 3 g 函数

$g(n,a,b,c) = \sum_{i=0}^{n} i \lfloor \frac{ai+b}{c} \rfloor$
当 $a \ge c$ 或 $b \ge c$ 的时候，跟 $f$ 函数的化简类似我们可以得到
$g(n,a,b,c) = \frac{n*(n+1)(2n+1)}{6} \lfloor \frac{a}{c} \rfloor + \frac{n*(n+1)}{2} \lfloor \frac{b}{c} \rfloor + g(n, a\%c, b\%c, c)$
当 $a < c$ 而且 $b < c$
$g(n,a,b,c) = \sum_{i=0}^{n} i \sum_{j=1}^{m} [j \le \lfloor \frac{ai+b}{c} \rfloor]$
$g(n,a,b,c) = \sum_{i=0}^{n} i \sum_{j=0}^{m-1} [\lfloor \frac{jc+c-b-1}{a} \rfloor < i]$
令 $t = \lfloor \frac{jc+c-b-1}{a} \rfloor$
那么
$g(n,a,b,c) = \sum_{j=0}^{m-1} \sum_{i=0}^{n} [i > t] i$
后面部分是一个连续的数列
$g(n,a,b,c) = \sum_{j=0}^{m-1} \frac{1}{2}(t+1+n)(n-t)$
$g(n,a,b,c) = \sum_{j=0}^{m-1} \frac{1}{2}(n^2+n-t^2-t)$
$g(n,a,b,c) = \frac{1}{2}(mn(n+1) - \sum_{j=0}^{m-1} t^2 - \sum_{j=0}^{m-1} t)$
$g(n,a,b,c) = \frac{1}{2}(mn(n+1) - h(m-1, c, c-b-1, a) - f(m-1, c, c-b-1, a))$

# 4 h 函数

h 函数的推导留给大家自己尝试，这里给出最终形式

推导过程注意一个性质的应用

$n^2 = 2\sum_{i=0}^{n} i - n$

这样子可以化简平方项

$h(n,a,b,c) = nm(m+1) - 2g(m-1,c,c-b-1,a) - 2f(m-1,c,c-b-1,a) - f(n,a,b,c)$

# 5 总结

现在，我们发现其实 g 和 h 函数的求解是三个函数交错递归的

实现的时候如果是暴力递归计算会重复计算，由于 $f,g,h$ 三个函数在两种条件下递归之后的参数是一样的，所以可以一起计算，回溯的时候再计算当前参数下对应的 $f,g,h$ 函数值

代码实现留给大家自己尝试，应该不难