# 斐波那契的最小公倍数 解题报告

安徽师范大学附属中学 吴作凡

## 1 试题来源

51nod的640分题，链接：https://www.51nod.com/question/index.html#!questionId=1223

## 2 试题大意

斐波那契数列定义如下：$F(0)=0, F(1)=1, F(n)=F(n-1)+F(n-2)$

给出$n$个正整数$a_1, a_2, ...a_n$，求$F(a_1), F(a_2), ..., F(a_n)$的最小公倍数，对1000000007取模。

数据范围：$n \leq 50000, a_i \leq 1000000$

## 3 算法介绍

要处理这个题我们先要发现斐波那契数列的一个性质：

$$gcd(F(a), F(b)) = F(gcd(a, b))$$

$gcd$表示最大公约数，接下来我们来证明一下这个性质。

首先我们可以证明$gcd(F(a), F(a+1)) = 1$，这个我们可以用数学归纳法证明：$gcd(F(0), F(1)) = 1$，而由于$gcd$的一些性质

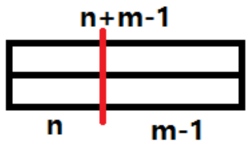
$$gcd(F(a), F(a+1)) = gcd(F(a), F(a) + F(a-1)) = gcd(F(a), F(a-1)) = 1$$

然后我们可以证明$F(n+1)F(m) + F(n)F(m-1) = F(n+m)$，这个式子用数学归纳法也很好证明，这里我们可以使用一种组合的方法证明。

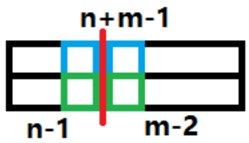
考虑用$1\times2$的多米诺骨牌覆盖一个$2\times n$的网格，设方案数是$f(n)$，考虑转移，如果最后一列放竖的骨牌那么方案数是$f(n-1)$，如果放两个横的就是$f(n-2)$，

于是$f(n) = f(n-1) + f(n-2)$，这个式子和斐波那契数列一模一样！当然我们需要考虑初值问题，可以发现$f(n) = F(n+1)$。

现在我们对$2 \times (n+m-1)$的网格进行覆盖，如果这种方案在$n$的位置我们可以将其切断，那么方案就是$f(n)f(m-1)$




否则在$n$和$n+1$一定是两个横过来的骨牌，所以方案数是$f(n-1)f(m-2)$




于是就有

$$f(n+m-1) = f(n)f(m-1)+f(n-1)f(m-2) = F(n+1)F(m)+F(n)F(m-1) = F(n+m)$$

于是我们在考虑归纳原来的式子，首先我们有$gcd(F(a), F(a+1)) = 1$，于是

$$\begin{aligned} gcd(F(a), F(b)) &= gcd(F(a), F(b-a+1)F(a) + F(b-a)F(a-1)) \\ &= gcd(F(a), F(b-a)F(a-1)) = gcd(F(a), F(b-a)) \end{aligned}$$

而$gcd(a, b) = gcd(a, b-a)$，这样就归纳成功了。

于是我们就能轻松地求出斐波那契的最大公约数了，而原题是最小公倍数，我们需要转化。

考虑每个质因子，$gcd$是对指数取min，而$lcm$是取max，这提醒我们可以用min-max容斥！所谓的min-max容斥就是对于一个数集$S$，我们有

$$\max\{S\} = \sum_{T \subseteq S, |T|>0} (-1)^{|T|-1} \min\{T\}$$

证明的话就只要考虑一个数会被当成哪些集合的min，发现只有max才会有一次贡献。

那么我们就得到了类似的容斥式子：对于一个数集$S$，我们有（用$lcm$表示最小公倍数）

$$lcm\{S\} = \prod_{T \subseteq S, |T|>0} gcd\{T\}^{(-1)^{|T|-1}}$$

我们就将$lcm$转化为了$gcd$。

于是答案就是

$$lcm(F(a_1), F(a_2)...) = F(a_1)F(a_2).../F(gcd(a_1, a_2))... \times F(gcd(a_1, a_2, a_3))...$$

那么我们就需要对于每个数$x$求出$F(x)$在答案中的指数$g(x)$，也就是它作为$gcd$的容斥系数和。

作为$gcd$的限制太苛刻，我们考虑让其作为$gcd$的因子，求出系数和$h(x)$，于是$h(x) = \sum_{x|d} g(d)$，可以通过对$h$反演得到$g$。

如果让$x$作为$gcd$的因子，我们就只需要求出有多少个$a$是$x$的倍数，可以发现如果个数不为0，那么$h(x) = 1$，否则$h(x) = 0$，那么$h(x)$很好算，枚举$x$的每个倍数判断是否出现过就好了。

我们设$a$的上限是$m$，于是求出$g$以后答案就是$\prod_{k=1}^{m} F(k)^{g(k)}$，直接统计就好。

时间复杂度$O(m \log m)$。