# 5.3 应用：求阶和因子问题——习题与详细解答

DanX, Joe Chen[*] and Li Fan

**练习 5.10**（（阶的计算示例））**.** *求 $x = 5$ 在模 $N = 21$ 意义下的阶 $r$。也就是说，找到最小的正整数 $r$，使得*

$$5^r \equiv 1 \pmod{21}.$$

**解答.** "阶"就是"把同一个数一直乘下去，多久乘回 1"。我们直接把 5 的幂在模 21 下算一算：

$$\begin{aligned}
&5^1 \equiv 5 \pmod{21},\\
&5^2 = 25 \equiv 4 \pmod{21},\\
&5^3 = 5^2 \cdot 5 \equiv 4 \cdot 5 = 20 \pmod{21},\\
&5^4 = 5^3 \cdot 5 \equiv 20 \cdot 5 = 100 \equiv 16 \pmod{21},\\
&5^5 = 5^4 \cdot 5 \equiv 16 \cdot 5 = 80 \equiv 17 \pmod{21},\\
&5^6 = 5^5 \cdot 5 \equiv 17 \cdot 5 = 85 \equiv 1 \pmod{21}.
\end{aligned}$$

我们看到第一个回到 1 的幂是 $r = 6$，而 $1 \le k \le 5$ 时 $5^k$ 都不等于 $1 \pmod{21}$。因此，

$$5 \text{ 在模 } 21 \text{ 下的阶} r = 6.$$

□

**练习 5.11**（（阶的上界））**.** *设 $x$ 与 $N$ 互质，即 $\gcd(x, N) = 1$，记 $x$ 在模 $N$ 意义下的阶为 $r$。证明 $r \leqslant N$。*

**解答.** 这里需要用到两个事实：

1. 若 $\gcd(x, N) = 1$，则存在正整数 $\varphi(N)$（Euler 函数），使得

$$x^{\varphi(N)} \equiv 1 \pmod{N}.$$

(这是 Euler 定理。)

2. 阶 $r$ 的定义：$x^r \equiv 1 \pmod{N}$，且 $r$ 是满足这个性质的最小正整数。

既然 $x^{\varphi(N)} \equiv 1 \pmod{N}$，说明 $\varphi(N)$ 也是一个"让 $x$ 回到 1 的幂数"。而 $r$ 是最小的那一个，所以必有

$$r \leqslant \varphi(N).$$

另一方面，Euler 函数总满足 $\varphi(N) \leqslant N - 1 < N$（因为 $1, \dots, N-1$ 中最多也就 $N-1$ 个数与 $N$ 互质）。于是

$$r \leqslant \varphi(N) < N,$$

从而 $r \leqslant N$ 证毕。

直观地说：在模 $N$ 的世界里，数的"周期"不会比整个世界的大小还大。 □

[*]qhc.statistics@gmail.com

**练习 5.12** (（模乘算符的酉性）). 设 $x$ 与 $N$ 互质，在 $N$ 维空间

$$\mathcal{H} = span\{|0\rangle, |1\rangle, \dots, |N-1\rangle\}$$

上定义线性算符

$$U|y\rangle = |xy \pmod N\rangle, \qquad y = 0, \dots, N-1.$$

证明 $U$ 是酉算符（即 $U^\dagger U = I$）。进一步写出 $U^{-1}$ 对基矢 $|y\rangle$ 的作用。

**解答. 1. 证明 $U$ 是酉的。**

判断一个算符是否酉，有几个等价的标准：

- 看它在这组基下的矩阵是否为“置换矩阵”（每列恰好有一个 1、其它 0）; - 或者验证

$$\langle y'|y\rangle = \langle y'|U^\dagger U|y\rangle \quad 对所有y, y'.$$

我们采用第二种做法。先算

$$\langle y'|U^\dagger U|y\rangle = \langle Uy'|\,Uy\rangle = \langle xy' \pmod N|xy \pmod N\rangle\,.$$

这就是问：在模 $N$ 下，$xy'$ 和 $xy$ 是否是同一个数。

因为 $\gcd(x, N) = 1$，乘以 $x$ 在模 $N$ 下是可逆的：如果

$$xy' \equiv xy \pmod N,$$

那就可以把两边同时乘上 $x$ 的模逆 $x^{-1}$，得到

$$y' \equiv y \pmod N.$$

而 $y, y'$ 都在 $0, \dots, N-1$ 这个范围内，因此只能是 $y' = y$。

于是：

$$xy' \equiv xy \pmod N \iff y' = y.$$

换到量子态的内积上，就是

$$\langle xy' \pmod N|xy \pmod N\rangle = \delta_{y',y}.$$

因此

$$\langle y'|U^\dagger U|y\rangle = \delta_{y',y},$$

说明 $U^\dagger U = I$，$U$ 是酉算符。

**2. $U^{-1}$ 的作用形式。**

因为 $U$ 是酉的，所以 $U^{-1} = U^\dagger$。从定义

$$U|y\rangle = |xy \pmod N\rangle$$

可以看出，$U^{-1}$ 的作用应该是“乘以 $x$ 的逆元”。

更具体些：设 $r$ 是 $x$ 在模 $N$ 下的阶，即

$$x^r \equiv 1 \pmod N.$$

于是

$$x^{r-1} \cdot x \equiv 1 \pmod N,$$

说明 $x^{r-1}$ 正是 $x$ 的模逆 $x^{-1}$。因此

$$U^{-1}|y\rangle = \left|x^{r-1}y \pmod N\right\rangle.$$

直观上，$U$ 是“把 $y$ 乘以 $x$”，$U^{-1}$ 就是“把 $y$ 乘以 $x^{r-1}$ 来抵消掉这个 $x$”。 □

**练习 5.13**（（本征态的离散 Fourier 展开））**.** 在书中，$U$ 的本征态 $|u_s\rangle$（$s=0,\ldots,r-1$）定义为

$$|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i s k/r}\left|x^k \pmod N\right\rangle.$$

证明

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{2\pi i s k/r}\,|u_s\rangle = \left|x^k \pmod N\right\rangle, \qquad k=0,\ldots,r-1. \tag{1}$$

进一步证明

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle.$$

提示：需要用到离散 *Fourier* 变换的正交恒等式

$$\sum_{s=0}^{r-1} e^{-2\pi i s k/r} = r\,\delta_{k0}.$$

**解答.** **1. 证明式** (1)。

把 $|u_s\rangle$ 的定义代入左边：

$$\begin{aligned}\text{LHS} &= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{2\pi i s k/r}\left[\frac{1}{\sqrt{r}}\sum_{k'=0}^{r-1} e^{-2\pi i s k'/r}\left|x^{k'}\right\rangle\right] \\ &= \frac{1}{r}\sum_{s=0}^{r-1}\sum_{k'=0}^{r-1} e^{2\pi i s (k-k')/r}\left|x^{k'}\right\rangle.\end{aligned}$$

交换求和顺序：

$$\text{LHS} = \frac{1}{r}\sum_{k'=0}^{r-1}\left[\sum_{s=0}^{r-1} e^{2\pi i s (k-k')/r}\right]\left|x^{k'}\right\rangle.$$

注意里面的和只是把 $k-k'$ 换个记号而已：

$$\sum_{s=0}^{r-1} e^{2\pi i s (k-k')/r} = \sum_{s=0}^{r-1} e^{-2\pi i s (k'-k)/r} = r\,\delta_{k,k'}.$$

于是

$$\text{LHS} = \frac{1}{r}\sum_{k'=0}^{r-1} r\,\delta_{k,k'}\left|x^{k'}\right\rangle = \left|x^k\right\rangle,$$

即得到 (1)。

**2. 推出** $\frac{1}{\sqrt{r}}\sum_s |u_s\rangle = |1\rangle$。

有两种等价的想法：

方法一：直接令 $k=0$。

在式 (1) 中令 $k=0$，注意 $x^0\equiv 1$，得到

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = \left|x^0\right\rangle = |1\rangle,$$

立刻完成。

方法二：利用 $U^m$ 作用在本征态上的性质。

我们也可以按照书中的提示，对 (1) 的两边同时作用 $U^{-k}$。

- 右边：$U^{-k}\left|x^k\right\rangle = |1\rangle$；- 左边：记本征值 $\lambda_s = e^{2\pi i s/r}$，则

$$U^{-k}\left|u_s\right\rangle = \lambda_s^{-k}\left|u_s\right\rangle .$$

因此

$$U^{-k}\left[\frac{1}{\sqrt{r}}\sum_s e^{2\pi i s k/r}\left|u_s\right\rangle\right] = \frac{1}{\sqrt{r}}\sum_s e^{2\pi i s k/r}\lambda_s^{-k}\left|u_s\right\rangle = \frac{1}{\sqrt{r}}\sum_s \left|u_s\right\rangle .$$

两边结果必须相等，所以

$$\frac{1}{\sqrt{r}}\sum_s \left|u_s\right\rangle = |1\rangle .$$

这个结论的直观含义是：把一组本征态按相同权重叠加起来，可以得到 $U$ 的某个“对称态”——在这里正好就是 $|1\rangle$。 □

**练习 5.14**（（另一种构造求阶电路的方式））**.** *若第二寄存器初始化为 $|1\rangle$，在逆 Fourier 变换之前，求阶算法产生的联合态为*

$$|\psi\rangle = \sum_{j=0}^{2^t-1}|j\rangle\, U^j\,|1\rangle = \sum_{j=0}^{2^t-1}|j\rangle\left|x^j \pmod N\right\rangle .$$

*证明：若把 $U^j$ 换成酉算符 $V$，其作用为*

$$V\,|j\rangle\,|k\rangle = |j\rangle\left|k + x^j \pmod N\right\rangle ,$$

*并让第二寄存器从 $|0\rangle$ 开始，也能得到同样的态。再说明如何仍旧用 $O(L^3)$ 个基本门（这里 $L$ 是 $N$ 的比特数）构造 $V$。*

**解答.** **1. 两种做法产生相同的态。**

现在第二寄存器初态换成 $|0\rangle$，并使用 $V$。一开始联合态为

$$\sum_{j=0}^{2^t-1}|j\rangle\,|0\rangle .$$

作用 $V$：

$$V\left(\sum_j |j\rangle\,|0\rangle\right) = \sum_j V\,|j\rangle\,|0\rangle = \sum_j |j\rangle\left|0 + x^j \pmod N\right\rangle = \sum_j |j\rangle\left|x^j \pmod N\right\rangle .$$

这与题目中原始做法的结果一模一样，所以“把 $U^j$ 全部记到一个加法里”是等价的。

**2. $V$ 的门复杂度。**

构造 $V$ 实际上要做两件事：

1. 由控制寄存器里的 $j$ 计算出 $x^j \pmod N$；2. 把这个值加到第二寄存器的 $k$ 上（模 $N$ 加法）。

在 Shor 算法里，我们本来就需要一个“模幂运算”电路来实现

$$|j\rangle\,|1\rangle \mapsto |j\rangle\left|x^j \pmod N\right\rangle$$

其复杂度已经分析过是 $O(L^3)$。在这里只是把输出不再直接作为第二寄存器的值，而是加到 $k$ 上：

- 模加法可以在 $O(L)$ 个基本门内完成；- 计算 $x^j \pmod N$ 本身仍然是 $O(L^3)$。

把两者加起来，复杂度仍然是 $O(L^3)$，只差了一个多项式里的低阶项。因此改用 $V$ 并没有让算法更慢。 □

**练习 5.15**（（最小公倍数与最大公因数））**.** 证明正整数 $x, y$ 的最小公倍数 $\operatorname{lcm}(x, y)$ 为

$$\operatorname{lcm}(x, y) = \frac{xy}{\gcd(x, y)},$$

其中 $\gcd(x, y)$ 是最大公因数。由此说明：若 $x, y$ 是 $L$ 比特的整数，那么可以在多项式时间内（例如 $O(L^2)$ 或 $O(L^3)$ 步）计算出它们的最小公倍数。

**解答.** **1. 证明公式** $\operatorname{lcm}(x, y) = xy/\gcd(x, y)$。

设

$$g = \gcd(x, y).$$

根据最大公因数定义，存在整数 $a, b$ 使得

$$x = ag, \qquad y = bg,$$

且 $\gcd(a, b) = 1$（否则公因数还可以进一步放大）。

记 $t = \operatorname{lcm}(x, y)$。按定义，$t$ 同时被 $x$ 和 $y$ 整除，所以

$$t = \alpha x = \beta y$$

对某些正整数 $\alpha, \beta$ 成立。代入 $x = ag, y = bg$：

$$t = \alpha ag = \beta bg.$$

两边除以 $g$ 得

$$\alpha a = \beta b.$$

因为 $a$ 和 $b$ 互质，$a$ 的所有质因数都不能出现在 $b$ 里，反之亦然。要让 $\alpha a = \beta b$ 成立，唯一的办法是

$$\alpha = b, \qquad \beta = a$$

（否则等式两边的质因子幂次对不上）。

于是

$$t = \alpha x = b \cdot ag = abg = \frac{ag \cdot bg}{g} = \frac{xy}{\gcd(x, y)}.$$

这就是想要的公式。

**2. 关于计算复杂度。**

要算 $\operatorname{lcm}(x, y)$，只需要做三步：

1. 用 Euclid 辗转相除法算出 $g = \gcd(x, y)$；2. 算出乘积 $xy$；3. 做一次整数除法 $(xy)/g$。

在常见的复杂度模型下，一次 $L$ 比特的乘法、除法或者取模都可以在 $O(L^2)$ 或更快的时间内完成；而 Euclid 算法需要做 $O(L)$ 次这样的操作，总复杂度大约是 $O(L^2)$ 到 $O(L^3)$ 之间，反正都是多项式级别。

因此，无论细节常数如何，求最小公倍数是一个经典多项式时间的问题，完全可以交给普通计算机去做。 □

**练习 5.16**（（一个关于素数和的估计））**.** 对所有 $x \geqslant 2$，证明

$$\int_x^{x+1} \frac{1}{y^2}\,\mathrm{d}y \geqslant \frac{2}{3x^2}.$$

进而证明

$$\sum_q \frac{1}{q^2} \leqslant \frac{3}{2}\int_2^{+\infty} \frac{1}{y^2}\,\mathrm{d}y = \frac{3}{4},$$

其中求和是对所有素数 $q$。由此得到书中式 *(5.58)* 中的概率下界

$$1-\sum_q p(q|s_1')p(q|s_2') \geqslant \frac{1}{4}.$$

**解答. 1. 计算并估计定积分。**

先直接算积分:

$$\int_x^{x+1} \frac{1}{y^2}\,\mathrm{d}y = \left[-\frac{1}{y}\right]_x^{x+1} = \frac{1}{x} - \frac{1}{x+1} = \frac{1}{x(x+1)}.$$

要证明

$$\frac{1}{x(x+1)} \geqslant \frac{2}{3x^2},$$

等价于

$$\frac{1}{x(x+1)} - \frac{2}{3x^2} \geqslant 0.$$

把两边都乘上正数 $3x^2(x+1)$:

$$3x^2 \geqslant 2x(x+1) \iff 3x^2 \geqslant 2x^2+2x \iff x^2 \geqslant 2x \iff x \geqslant 2.$$

这正是题设条件，所以不等式成立。也可以把结果改写为

$$\frac{1}{x^2} \leqslant \frac{3}{2}\int_x^{x+1} \frac{1}{y^2}\,\mathrm{d}y.$$

**2. 用积分来估计 $\sum_q 1/q^2$。**

记住：素数集合是 $\{2,3,5,7,\dots\}$，它是 $\{2,3,4,5,\dots\}$ 的子集，所以

$$\sum_q \frac{1}{q^2} < \sum_{x=2}^{\infty} \frac{1}{x^2}.$$

对每个 $x \geqslant 2$，由上面的不等式有

$$\frac{1}{x^2} \leqslant \frac{3}{2}\int_x^{x+1} \frac{1}{y^2}\,\mathrm{d}y.$$

于是

$$\sum_{x=2}^{\infty} \frac{1}{x^2} \leqslant \frac{3}{2}\sum_{x=2}^{\infty}\int_x^{x+1} \frac{1}{y^2}\,\mathrm{d}y = \frac{3}{2}\int_2^{+\infty} \frac{1}{y^2}\,\mathrm{d}y.$$

最后一个积分很好算:

$$\int_2^{+\infty} \frac{1}{y^2}\,\mathrm{d}y = \left[-\frac{1}{y}\right]_2^{+\infty} = \frac{1}{2}.$$

因此

$$\sum_q \frac{1}{q^2} < \sum_{x=2}^{\infty} \frac{1}{x^2} \leqslant \frac{3}{2}\cdot\frac{1}{2} = \frac{3}{4}.$$

**3. 回到概率下界。**

书中有

$$1-\sum_q p(q|s_1')p(q|s_2') \geqslant 1-\sum_q \frac{1}{q^2}.$$

上面我们已经证明 $\sum_q 1/q^2 \leqslant 3/4$，所以

$$1 - \sum_q p(q|s_1')p(q|s_2') \geqslant 1 - \frac{3}{4} = \frac{1}{4}.$$

这说明：在算法里重复做两次相位估计，并通过取最大公因数来恢复阶的做法，成功概率至少有 25%，实际上还会更高。 □

**练习 5.17**（（判断一个数是不是整数幂））**.** *设 $N$ 是一个 $L$ 比特的正整数。本题希望给出一个高效的经典算法，判断是否存在整数 $a > 1$、$b \geqslant 2$ 使得*

$$N = a^b.$$

*提示步骤如下：*

1. *证明若这样的 $b$ 存在，则必有 $b \leqslant L$；*
2. *说明：计算 $y = \log_2 N$，对所有 $b \leqslant L$ 计算 $x = y/b$，再计算最接近 $2^x$ 的两个整数 $u_1, u_2$，总共只需 $O(L^2)$ 次基本运算；*
3. *说明：计算 $u_1^b, u_2^b$ 并检查是否等于 $N$，只需 $O(L^2)$ 次基本运算；*
4. *综合以上结论，给出一个 $O(L^3)$ 的判定算法。*

**解答.** 整个思路：枚举可能的指数 $b$，对每个 $b$ 找出"最有可能的底数"候选，然后检验是否真能得到 $N$。

**(1)** $b \leqslant L$。

若 $N = a^b$，且 $a > 1$，那么 $a \geqslant 2$。于是

$$N = a^b \geqslant 2^b.$$

两边取 $\log_2$：

$$\log_2 N \geqslant b.$$

又因为 $N$ 是 $L$ 比特数，有 $2^{L-1} \leqslant N < 2^L$，所以

$$L - 1 \leqslant \log_2 N < L.$$

综合起来：

$$b \leqslant \log_2 N < L,$$

所以可以简单地认为 $b \leqslant L$。这意味着我们只需要枚举 $b = 2, 3, \dots, L$ 这么多种可能即可，循环次数是 $O(L)$。

**(2) 粗略地找出底数候选**。

设 $y = \log_2 N$，如果 $N = a^b$ 成立，那么

$$\log_2 N = \log_2(a^b) = b \log_2 a.$$

于是

$$x := \frac{y}{b} = \log_2 a, \qquad a = 2^x.$$

因此对给定的 $b$，$a$ 应该接近 $2^x$。我们不一定能算出 $2^x$ 的精确整数值，但可以算出一个浮点近似，再向下、向上各取一个最接近的整数 $u_1 = \lfloor 2^x \rfloor$、$u_2 = \lceil 2^x \rceil$ 作为候选。真正的 $a$ 如果存在，一定在这两个数里。

在复杂度层面上：

- 计算一次 $y = \log_2 N$，可以看成对一个 $L$ 比特数做一些标准运算（如逐位处理），复杂度在 $O(L^2)$ 左右；- 对每个 $b$，算 $x = y/b$ 是一次除法；- 再算 $2^x$（实数指数）并四舍五入得到 $u_1, u_2$，可以用标准的数值算法实现，其核心也是对 $L$ 比特数做若干次加减乘除，复杂度在 $O(L^2)$ 的量级。

我们只需要把这一点记住：对单个 $b$ 做这些操作是 $O(L^2)$ 的。

**(3) 检查** $u_1^b, u_2^b$。

接下来，对这两个候选底数 $u_1, u_2$，需要检验

$$u_1^b \stackrel{?}{=} N, \qquad u_2^b \stackrel{?}{=} N.$$

计算幂可以用“反复平方法”或“反复乘法”：

- 例如要算 $u^b$，可以从 $u^1$ 开始，每次再乘一个 $u$，一共乘 $b-1$ 次；- 在第 2 步中我们是对不同的 $b$ 进行循环，可以把上一次计算的结果存下来，用一次乘法更新到下一次需要的幂次。

每次乘法作用在 $L$ 比特整数上，复杂度是 $O(L^2)$；幂的计算最多也就做常数次（因为候选只有两个），加上和 $N$ 比较，也都在 $O(L^2)$ 以内。

**(4) 总体算法与复杂度**。

综上，可以设计下列算法：

1. 先计算 $y = \log_2 N$；
2. 对 $b = 2, 3, \ldots, L$ 依次执行：
   (a) 计算 $x = y/b$；
   (b) 找到最接近 $2^x$ 的两个整数 $u_1, u_2$；
   (c) 计算 $u_1^b, u_2^b$ 并与 $N$ 比较；
   (d) 若发现 $u_i^b = N$，就找到了 $N = a^b$ 的表示；否则继续下一个 $b$。

对每个 $b$，内部的运算量是 $O(L^2)$，而 $b$ 的取值个数是 $O(L)$，所以总复杂度是

$$O(L) \times O(L^2) = O(L^3).$$

如果循环结束仍然没有找到任何一对 $(a, b)$ 满足 $N = a^b$，就可以断定 $N$ 不是整数幂。这样就给出了一个经典的 $O(L^3)$ 判定算法。 □

**练习 5.18**（（因式分解 91 的示例））**.** *按 Shor 算法中的“因式分解 ⇒ 求阶”归约，考虑* $N = 91$。

1. *验证第 1 步和第 2 步不会提前退出：即 91 不是偶数，也不是更小整数的幂；*
2. *在第 3 步中选择* $x = 4$ *（与 91 互质），计算* $x$ *在模* $N$ *下的阶* $r$；
3. *证明* $r$ *为偶数，且* $x^{r/2} \not\equiv -1 \pmod{91}$，*从而算法会成功地给出一个非平凡因子，并算出这个因子。*

**解答.** **1. 前两步检查。**

- 第 1 步：91 不是偶数，所以不会直接返回因子 2；- 第 2 步：$91 = 7 \cdot 13$，不是 $a^b$ 的形式（$3^2 = 9$、$4^2 = 16$、$5^2 = 25$、$6^2 = 36$、$7^2 = 49$、$8^2 = 64$、$9^2 = 81$，都不等于 91），所以也不会在这一步退出。

因此算法会进入第 3 步选取随机的 $x$。

**2. 计算 $x = 4$ 的阶。**

检查 $\gcd(4, 91) = 1$（显然），于是可以继续求阶。我们依次计算 $4^k \bmod 91$：

$$\begin{aligned}
4^1 &= 4 \equiv 4 \pmod{91},\\
4^2 &= 16 \equiv 16 \pmod{91},\\
4^3 &= 64 \equiv 64 \pmod{91},\\
4^4 &= 64 \cdot 4 = 256 \equiv 256 - 2 \cdot 91 = 74 \pmod{91},\\
4^5 &= 74 \cdot 4 = 296 \equiv 296 - 3 \cdot 91 = 23 \pmod{91},\\
4^6 &= 23 \cdot 4 = 92 \equiv 1 \pmod{91}.
\end{aligned}$$

第一次回到 1 的幂是 $k = 6$，所以阶 $r = 6$。

**3. 检查 $x^{r/2}$ 并求因子。**

阶为 $r = 6$，是偶数。按照 Shor 算法下一步需要检查

$$x^{r/2} = 4^3 = 64 \pmod{91}$$

是否等于 $-1 \pmod{91}$。而

$$-1 \pmod{91} \equiv 90,$$

显然 $64 \neq 90$，所以 $4^{r/2} \not\equiv -1 \pmod{91}$，算法可以继续。

接下来计算

$$\gcd(4^{r/2} - 1, 91) = \gcd(64 - 1, 91) = \gcd(63, 91).$$

可以用辗转相除法：

$$91 = 1 \cdot 63 + 28, \qquad 63 = 2 \cdot 28 + 7, \qquad 28 = 4 \cdot 7 + 0,$$

所以 $\gcd(63, 91) = 7$。

同样地，

$$\gcd(4^{r/2} + 1, 91) = \gcd(65, 91) = 13.$$

于是我们得到了 91 的两个非平凡因子 7 和 13，因式分解完成。

**小结：** 这一题展示了 Shor 算法中“求阶 $\Rightarrow$ 求因子”的核心步骤在一个小例子上的具体操作。真正的困难其实在于高效求阶，这正是量子部分的工作。 □

**练习 5.19**（（最小的需要求阶的合数））**.** 证明：$N = 15$ 是最小的、确实需要用到“求阶子程序”的合数。更准确地说，它是最小的非偶数、且不是更小正整数幂的合数。

**解答.** 按照书中“因式分解 $\Rightarrow$ 求阶”的经典预处理步骤：

1. 如果 $N$ 是偶数，那么直接得到因子 2，无需求阶；
2. 如果 $N$ 可以写成 $a^b$（$b \geqslant 2$），同样存在一些简单方法分解它，也不需要求阶；

3. 只有当 $N$ 既不是偶数、又不是更小整数幂时，才真正进入求阶阶段。

现在枚举最小的合数：

$$4, 6, 8, 9, 10, 12, 14, 15, \ldots$$

逐个检查：

- $4 = 2^2$，是偶数也是整数幂，早就会被前两步排除；- $6, 8, 10, 12, 14$ 都是偶数，会在第 1 步直接得到因子 2；- $9 = 3^2$ 是奇数，但它是整数幂，会在第 2 步被识别出来。

因此，在 $4, 6, 8, 9, 10, 12, 14$ 之后，第一个既不是偶数又不是整数幂的合数就是

$$15 = 3 \times 5.$$

这就说明：*15 是最小的"需要真正调用求阶子程序"的合数*。这也是为什么很多入门教材都会用 15 作为 Shor 算法的第一个完整示例。 □