4.3.7 欧几里得算法

直接从整数的素因子分解式计算两个整数的最大公约数是效率很低的。原因是寻找素因子分解式非常耗时。这里给出一个更高效的寻找最大公约数的方法，称为**欧几里得算法**。这个算法古代就有了。这是用古希腊数学家欧几里得的名字命名的，他在其著作《几何原本》(The Elements)中记载了这一算法的描述。

在介绍欧几里得算法之前，我们先看一看它是怎样求 gcd(91，287)的。首先，用两个数中的大数 287 除以两个数中的小数 91，得到

$$287 = 91 \cdot 3 + 14$$

91 和 287 的任何公约数必定也是 $287 - 91 \cdot 3 = 14$ 的因子。而且 91 和 14 的任何公约数也必定是 $287 = 91 \cdot 3 + 14$ 的因子。因此，287 和 91 的最大公约数和 91 与 14 的最大公约数相同。这意味着求 gcd(91，287)的问题已被归约为求 gcd(91，14)的问题。

接下来，91 除以 14 得

$$91 = 14 \cdot 6 + 7$$

由于 91 和 14 的任何公约数也能整除 $91 - 14 \cdot 6 = 7$，并且 14 和 7 的任何公约数整除 91，所以 gcd(91，14)＝gcd(14，7)。

继续 14 除以 7，得

$$14 = 7 \cdot 2$$

因为 7 整除 14，所以 gcd(14，7)＝7。另外，因为 gcd(287，91)＝gcd(91，14)＝gcd(14，7)＝7，所以最初的问题得解。

现在介绍欧几里得算法在一般情况下是如何工作。我们将用辗转相除法把求两个正整数最大公约数的问题归约为求两个较小整数的最大公约数问题，直到两个整数之一为 0。

欧几里得算法的基础是下面关于最大公约数和整除算法的结论。

引理 1 令 $a = bq + r$，其中 a，b，q 和 r 均为整数。则 $\gcd(a, b) = \gcd(b, r)$。

证明 如果能证明 a 与 b 的公约数和 b 与 r 的公约数相同，也就证明了 $\gcd(a, b) = \gcd(b, r)$，因为这两对整数必定有相同的最大公约数。

因此，假定 d 整除 a 和 b。则可得 d 也整除 $a - bq = r$(根据 4.1 节定理 1)。因此，a 和 b 的任何公约数也是 b 和 r 的公约数。

类似地，假定 d 整除 b 和 r。则 d 也整除 $bq + r = a$。因此，b 和 r 的任何公约数也是 a 和 b 的公约数。

因此，$\gcd(a, b) = \gcd(b, r)$。 ◀

假定 a 和 b 为正整数，且 $a \geqslant b$。令 $r_0 = a$ 和 $r_1 = b$。当连续应用整除算法时，可得

$$\begin{aligned} r_0 &= r_1 q_1 + r_2 \quad 0 \leqslant r_2 < r_1 \\ r_1 &= r_2 q_2 + r_3 \quad 0 \leqslant r_3 < r_2 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n \quad 0 \leqslant r_n < r_{n-1} \\ r_{n-1} &= r_n q_n \end{aligned}$$

最终在这一辗转相除序列中会出现余数为 0，因为在余数序列 $a = r_0 > r_1 > r_2 > \cdots \geqslant 0$ 中至多包含 a 项。再者，从引理 1 可知

$$\begin{aligned} \gcd(a,b) &= \gcd(r_0,r_1) = \gcd(r_1,r_2) = \cdots = \gcd(r_{n-2},r_{n-1}) \\ &= \gcd(r_{n-1},r_n) = \gcd(r_n,0) = r_n \end{aligned}$$

因此，最大公约数是除法序列中最后一个非零余数。

例 16 用欧几里得算法寻找 414 和 662 的最大公约数。

解 连续相除得出：

$$\begin{aligned} 662 &= 414 \cdot 1 + 248 \\ 414 &= 248 \cdot 1 + 166 \\ 248 &= 166 \cdot 1 + 82 \\ 166 &= 82 \cdot 2 + 2 \\ 82 &= 2 \cdot 41。 \end{aligned}$$

因此，gcd(414，662)＝2，因为 2 是最后一个非零余数。 ◀

我们可以用右表格来总结这些步骤。

j	r_j	r_{j+1}	q_{j+1}	r_{j+2}
0	662	414	1	248
1	414	248	1	166
2	248	166	1	82
3	166	82	2	2
4	82	2	41	0

欧几里得算法用伪代码表示如算法 1 所示。

算法 1 欧几里得算法

```
procedure gcd(a，b：正整数)
x := a
y := b
while y≠0
    r := x mod y
    x := y
    y := r
return x{gcd(a，b)是 x}
```

在算法 1 中，x 和 y 的初值分别是 a 和 b。在过程的每一步，x 取 y 的值，而 y 取 x **mod** y 的值，即 x 除以 y 的余数。只要 $y \neq 0$，该过程就不断重复。当 $y = 0$ 时算法终止，而此时 x 的值，该过程中最后一个非零余数，为 a 和 b 的最大公约数。

我们将在 5.3 节研究欧几里得算法的时间复杂度，并证明求 a 和 b 的最大公约数所要的除法次数当 $a \geqslant b$ 时为 $O(\log b)$。