最大公因子(Highest Common Factor)与辗转相除法

求最大公因子的主要方法

求两个整数最大公约数主要的方法：

- 穷举法：分别列出两整数的所有约数，并找出最大的公约数。
- 素因数分解：分别列出两数的素因数分解式，并计算共同项的乘积。
- 短除法：两数除以其共同素因数，直到两数互素时，所有除数的乘积即为最大公约数。
- 辗转相除法：两数相除，取余数重复进行相除，直到余数为0时，前一个除数即为最大公约数。

计算过程

计算过程 [编辑]

辗转相除法是一种递归算法，每一步计算的输出值就是下一步计算时的输入值。[20]设k表示步骤数（从0开始计数），算法的计算过程如下。

每一步的输入是都是前两次计算的非负余数r_{k-1}和r_{k-2}。因为每一步计算出的余数都在不断减小，所以，r_{k-1}小于r_{k-2}。在第k步中，算法计算出满足以下等式的商q_k和余数 r_k：

$$r_{k-2} = q_k r_{k-1} + r_k$$

其中$0 \le r_k < r_{k-1}$。也就是r_{k-2}要不断减去r_{k-1}直到比r_{k-1}小。

为求简明，以下只说明如何求两个非负整数a和b的最大公约数（负数的情况是简单的）。在第一步计算时（$k = 0$），设r_{-2}和r_{-1}分别等于a和b，第2步（此时$k = 1$）时计算r_{-1}（即b）和r_0（第一步计算产生的余数）相除产生的商和余数，以此类推。整个算法可以用如下等式表示：

$$a = q_0 b + r_0$$
$$b = q_1 r_0 + r_1$$
$$r_0 = q_2 r_1 + r_2$$
$$r_1 = q_3 r_2 + r_3$$
$$\cdots$$

如果有$a < b$，算法的第一步实际上会把两个数字交换，因为这时a除以b所得的商q_0会等于0，余数r_0则等于a。然后，算法的第二步便是把b除以a，再计算所得之商和余数。所以，对于$k \ge 0$总有$r_k < r_{k-1}$，即运算的每一步中得出的余数一定小于上一步计算的余数。

由于每一步的余数都在减小并且不为负数，必然存在第N步时r_N等于0，使算法终止[21]，r_{N-1}就是a和b的最大公约数。其中N不可能无穷大，因为在r_0和0之间只有有限个自然数。

正确性证明

正确性的证明 [编辑]

辗转相除法的正确性可以分成两步来证明。[20]在第一步，我们会证明算法的最终结果r_{N-1}同时整除a和b。因为它是一个公约数，所以必然小于或者等于最大公约数g。在第二步，我们证明g能整除r_{N-1}。所以g一定小于或等于r_{N-1}。两个不等式只在$r_{N-1} = g$时同时成立。具体证明如下：

1. 证明r_{N-1}同时整除a和b：

   因为余数r_N是0，r_{N-1}能够整除r_{N-2}：

   $$r_{N-2} = q_N r_{N-1}$$

   因为r_{N-1}能够整除r_{N-2}，所以也能够整除r_{N-3}：

   $$r_{N-3} = q_{N-1} r_{N-2} + r_{N-1}$$

   同理可证r_{N-1}可以整除所有之前步骤的余数，包括a和b，即r_{N-1}是a和b的公约数，$r_{N-1} \le g$。

2. 证明最大公约数g能整除r_{N-1}：

   根据定义，a和b可以写成g的倍数：$a = mg$、$b = ng$，其中m和n是自然数。因为
   $r_0 = a - q_0 b = mg - q_0 ng = (m - q_0 n)g$，所以$g$整除$r_0$。同理可证$g$整除每个余数$r_1$, r_2, ..., r_{N-1}。因为最大公约数g整除r_{N-1}，因而$g \le r_{N-1}$。

因为第一步的证明告诉我们$r_{N-1} \le g$，所以$g = r_{N-1}$。即：[22][23]

$$g = \mathrm{GCD}(a, b) = \mathrm{GCD}(b, r_0) = \mathrm{GCD}(r_0, r_1) = \ldots = \mathrm{GCD}(r_{N-2}, r_{N-1}) = r_{N-1}$$

举例

举例里的，图形演示非常好，非常易于理解。

举例 [编辑]

例如，计算a = 1071和b = 462的最大公约数的过程如下：从1071中不断减去462直到小于462（可以减2次，即商q_0 = 2），余数是147：

$$1071 = 2 \times 462 + 147.$$

然后从462中不断减去147直到小于147（可以减3次，即q_1 = 3），余数是21：

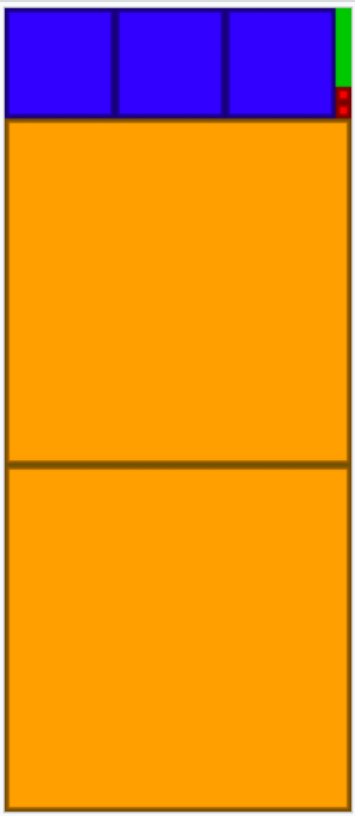
$$462 = 3 \times 147 + 21.$$

再从147中不断减去21直到小于21（可以减7次，即q_2 = 7），没有余数：

$$147 = 7 \times 21 + 0.$$

此时，余数是0，所以1071和462的最大公约数是21，这和用素因数分解得出的结果相同（见上文）用表格表示如下：

步骤数	算式	商和余数
0	$1071 = 462\ q_0 + r_0$	$q_0 = 2$、$r_0 = 147$
1	$462 = 147\ q_1 + r_1$	$q_1 = 3$、$r_1 = 21$
2	$147 = 21\ q_2 + r_2$	$q_2 = 7$、$r_2 = 0$（算法终止）



算法的演示动画。最初的绿色矩形的长和宽分别是a = 1071、b = 462，从中不断铺上462×462的正方形直到剩下部分面积是462×147；然后再铺上147×147的正方形直至剩下21×147的面积；最后，铺上21×21的正方形时，绿色部分就没有了。即21是1071和462的最大公约数。

图形演示 [编辑]

辗转相除法的计算过程可以用图形演示。[24]假设我们要在$a \times b$的矩形地面上铺正方形瓷砖，并且正好铺满，其中a大于b。我们先尝试用$b \times b$的瓷砖，但是留下了$r_0 \times b$的部分，其中$r_0 < b$。我们接着尝试用$r_0 \times r_0$的正方形瓷砖铺，又留下了$r_1 \times r_0$的部分，然后再使用$r_1 \times r_1$的正方形铺……直到全部铺满为止，即到某步时正方形刚好覆盖剩余的面积为止。此时用到的最小的正方形的边长就是原来矩形的两条边长的最大公约数。在图中，最小的正方形面积是21×21（红色），而原先的矩形（绿色）边长是1071×462，所以21是1071和462的最大公约数。

Show me the code

```
int GCD(int a, int b) {
    return a % b == 0 ? b : GCD(b, a % b);
}
```

参考

辗转相除法