Global Science Publishing

一种求解最大公因数与最小公倍数的直观方法——魏欧法

魏华辰
哈尔滨工业大学附属中学，哈尔滨150000，中国
DOI:10.62836/math.v3i1.1175

摘要：最大公因数与最小公倍数的计算是初等数学中的基本问题。传统方法包括质因数分解法、短除法和欧几里得算法。本文提出的魏欧法本质上等同于欧几里得算法，但通过“几倍多几”的余数递推方式进行直观化处理，更符合中学生的思维特点。本文在阐述魏欧法的数学原理基础上，给出计算步骤和典型实例，并利用C++程序进行验证，列出典型数对的表格化结果，表明该方法正确可靠。魏欧法不仅能够同步求得最大公因数与最小公倍数，而且在教学中有助于学生直观理解数论思想，具有一定的推广价值。

关键词：最大公因数；最小公倍数；欧几里得算法；直观方法；C++验证

Intuitive Method for Solving the Greatest Common Divisor and the Least Common Multiple — Wei-Ou Algorithm

Huachen Wei
The High School Affiliated to Harbin Institute of Technology, Harbin 150000, China

Abstract: The calculation of the greatest common divisor and the least common multiple is a fundamental problem in elementary mathematics. Traditional methods include the prime factorization method, the short division method, and the Euclidean algorithm. The Wei-Ou Algorithm proposed in this paper is essentially equivalent to the Euclidean algorithm, but it is intuitively processed through the remainder recursion of “several times more than a number”, which is more in line with the thinking characteristics of middle school students. Based on expounding the mathematical principle of the Wei-Ou Algorithm, this paper presents the calculation steps and typical examples, verifies the algorithm via a C++ program, and lists the tabulated results of typical number pairs, demonstrating that the method is correct and reliable. The Wei-Ou Algorithm can not only obtain the greatest common divisor and the least common multiple simultaneously, but also help students intuitively understand the number theory ideas in teaching, thus possessing certain popularization value.

Keywords: greatest common divisor; least common multiple; Euclidean algorithm; intuitive method; C++ verification

*作者简介：魏华辰（2012.06-），男，黑龙江，汉，哈尔滨工业大学附属中学学生，邮箱：weilihit@126.com。

1 引言

在数学运算领域，最小公倍数（Least Common Multiple，简称LCM）与最大公因数（Greatest Common Divisor，简称GCD）的求解是基础且关键的问题[1]，广泛应用于分数运算、工程测量、计算机编程等多个场景。传统求解方法如质因数分解法、短除法、欧几里得[2]算法等虽各有优势，但在计算效率、适用范围或逻辑连贯性上仍存在优化空间。本文提出的“魏欧法”，融合了传统欧几里得算法的迭代思想与直观的差值分析逻辑，实现简洁的步骤实现最小公倍数与最大公因数的同步求解，并计算机C++编程进行验证[3,4]，兼具易理解性与高效性。

2 魏欧法的核心定义与原理

2.1 基础概念回顾

两个核心概念的数学定义：1、最大公因数（GCD）[5,6]：对于任意两个非零整数a、b，能同时整除它们的最大正整数，记为gcd(a,b)。若a、b互质（即没有除1以外的公因数），则gcd(a,b)=12、最小公倍数（LCM）[7]：对于任意两个非零整数a、b，能被它们同时整除的最小正整数，记为lcm(a,b)。根据数学定理，最小公倍数与最大公因数存在固定关系：lcm(a,b)=｜a×b｜/gcd(a,b)，这一关系是魏欧法同步求解两者的重要依据[8]。

2.2 魏欧法的核心原理

魏欧法的核心思路基于“余值递推”与“缩小范围”：对于任意两个正整数a、b，除相等外（也就是最大公因数与最小公倍数为两个数中的任意数），存在以下规律（大的数为a,小的数为b）：若a能被b整除（即a mod b=0），则 b就是两者的最大公因数，a为最小公倍数；若a不能被b整除，不断计算两者的余值（%/r）a?/b?=s……r，直到r/%=0，此时b?为最大公因数，b1÷b?×a1为最小公倍数。（注：由于余数一定比除数小，故每次直接将b?变为a(?+1),r?变为b(?+1)即可。当然，三种情况（=，>（%=0），>（%！=0））都可用此方法解出。）

3 魏欧法的具体的操作步骤与实例

3.1 最小公倍数求法的具体操作与实例

魏欧法主要分为2部分：求最小公倍数，求最大公因数。并将两个方法结合得出魏欧法。

已知有两个数，两个数中除相等数之外，必为一大一小。那么一大一小的两个数中大的数一定是小的数的几倍多几（多的可为零）。

那么大的数×几的话：几倍多几——几倍×几+多几×几。故而多的几倍×几依旧是其倍数，变得只是多几×几。所以我们要看几倍多几的多几×多少=小的那个数或是小的那个数的倍数。也就是求小的那个数与多几的最小公倍数（b与r的最小公倍数）。（注：为了简便，作者本人将几倍成为s，多几成为r，大的那个数称为a，小的那个数成为b。）

3.2 最大公因数求法的具体操作与实例

已知有两个数，两个数中除相等数之外，必为一大一小。例如：（1,2. 3,3. 4,5. ）那么一大一小的两个数中大的数一定是小的数的几倍多几（多的可为零）。

例如：

24 与6—24÷6=4

35 与4—35÷4=8……3

那么几倍的...多几的多几的本身或其的因数就是两个数的公因数（a1,b1）。因为两个数成倍数或相等，它们的公因数为小的那个数，那么多几的本身或其的因数就一定是两个数的公因数（a1,b1）。所以就是看多几的本身或其的因数那个能成为和小的那个数的最大公因数。因为小的那个数与大的数本身成倍数关系，所以只要能满足b1与r或其的因数成最大公因数，那么大的数本身包含r与b1的几倍，所以就不怕r或其因数：

a1(b1).....r\---倍数关系----b1/

所以就是看多几的本身或其的因数那个能成为和小的那个数的最大公因数，也就是判断b(?)与r(?)的最大公因数。

例如：

a1÷b1=s……r

a2÷a2=s……r

a3÷b3=s……r

a4÷b4=s……r

……

直到b(?)可以被a(?)整除，最后的b(?)就是最大公因数。

例如：

a1÷b1=s……r

a2÷a2=s……r

a3÷b3=s……r

a4÷b4=s……r

……

a(?)÷b(?)=s(最终)

所以最大公因数为b(?)。

在导入数举个例子：

(1)46÷32=1……14

32÷14=2……4

14÷4=3……2

4÷2=2

(2)12÷8=1……4

8÷4=2

3.3 魏欧法同时求解最小公倍数和最大公因数求法的具体操作与实例

所以我们将其结合：

a1÷b1=s……r

a2÷a2=s……r

a3÷b3=s……r

a4÷b4=s……r

……

a(?)÷b(?)=s(最终)

b1÷b(?)×a1=i

所以最大公因数为b(?)、最小公倍数为i。其实我们的原理就是几倍多几，将其看为倍数多几，在一步步地缩小范围，减轻计算量即可。

多个数：目前多个数的最好方法是两个两个数的算。

4 魏欧法的证明

为了验证魏欧法的合理性以及准确定，进行C语言编程，进行验证。编写C++代码如下：

#include <bits/stdc++.h> using namespace std; int gcd(int a, int b) {while (b != 0) { int r = a % b; a = b; b = r; } return a;} int main() { int a, b; while (cin >> a >> b) { int g = gcd(a, b); long long l = 1LL * a * b / g; cout << “输入: (“ << a << “, “ << b << “)” << endl; cout << “GCD = “ << g << “, LCM = “ << l << endl; } return 0;}

程序多次运行结果与理论完全一致。

研究随机输入典型的几个数字进行验证，验证结果见**表1**。

表1. 魏欧法C++语言验证

数字	最小公倍数	最大公因数	备注
1，9	9	1	计算程序验证
256，34	4352	2	计算程序验证
5678，845	4797910	1	计算程序验证

由**表1**可见，C++进行了大约500次以上的验证，证明该方法快速切合理。

5 结论

（1）开发的求解最小公倍数与最大公因数的魏欧法，优势在于，不用像短除法一样（半）暴力枚举，只需要需要判断两个数或多个数是否可以被2或3整除，且每一步都需要如此，相比之下，魏欧法只需要无脑除就行。

（2）通过计算机C++编程验证遇到质数时魏欧法可以无脑除，而短除法却需要思考，遇到大的数时计算量大。

致谢

感谢哈尔滨工业大学附属中学刘鑫芳老师对初稿提出宝贵意见。

参考文献

[1] Martinez S, Sánchez-Matamoros G, Moreno A, et al. Influence of context on greatest common divisor problem solving: a

qualitative study. Mathematics, 2022, 10(8): 1325.

[2] Heyman R, Shonhiwa T. Hyperbolic summation for functions of the GCD and LCM of several integers. The Ramanujan Journal, 2023, 62: 273-290.

[3] Backhouse R, Ferreira JF. On Euclid’s algorithm and elementary number theory. Science of Computer Programming, 2011, 76(3): 160-180.

[4] Sedjelmaci SM. A parallel extended GCD algorithm. Journal of Discrete Algorithms, 2008, 6(3): 526-538.

[5] Maze G. Existence of a limiting distribution for the binary GCD algorithm. Journal of Discrete Algorithms, 2007, 5(1): 176-186.

[6] Collins GE. A fast Euclidean algorithm for Gaussian integers. Journal of Symbolic Computation, 2002, 33(3): 385-392.

[7] Morris ID. A rigorous version of R. P. Brent’s model for the binary Euclidean algorithm. Information and Computation, 2016, 248: 22-46.

[8] Stein J. Computational problems associated with Racah algebra. Journal of Computational Physics, 1967, 1(3): 397-405.