第34卷第1期　　中国科学技术大学学报　　Vol. 34, No. 1
2004年2月　JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA　Feb. 2004

文章编号:0253-2778(2004)01-0126-02

关于二进制 GCD 算法的注记*

孙翠芳

(安徽师范大学数学系,芜湖 241000)

关键词:最大公因子(gcd); Euclid 算法; 二进制 gcd 算法
中图分类号:TP301　　**文献标识码**:A
AMS Subject Classification(2000): Primary 11Y16, 11A05

求两个正整数 a、b 的最大公因子 gcd(a,b) 通常使用经典的 Euclid 算法. 因共需 $O(\ln N)$ 次带余除法,每次带余除法耗时 $O(\ln^2 N)$, 所以 Euclid 算法耗时 $O(\ln^3 N)$,这里 $N = \max(a,b)$, 文献[1, Corollary 2.1]和[2,例5]就是这样粗略估算的. 然而,如果在实现算法时考虑到每步带余除法被除数的位数在不断下降,总运行时间将仅为 $O(\ln^2 N)$, 文献[3, p. 328]和文献[4, p. 13]指出并证明了这一点,在文献[5]定理1的证明中也提到了这个事实.

1961 年 Stein 发明了一种求 gcd 的新算法(见[J. Comp. Phys. 1(1967),397 – 405]),简称"二进制 gcd 算法(Binary GCD)",它只需减法、奇偶校验和二进制位移运算(求偶数的一半),时间复杂度为 $O(\ln^2 N)$. 在多种教科书中,例如在被国际计算机科学界誉为计算机科学圣经的文献[3, p. 321], Springer 数学系列研究生教材[4, p. 14](GTM138)以及 文献[6, p. 132]中都详细介绍了这个算法并分析了其时间复杂度.

文献[7]又一次"提出了一个基于二进制的、适用于多重精度的、时间复杂度为 $O(\ln^2 N)$ 的改进算法",特别在§4 称其为"新"算法. 但事实上其数学原理与 Stein 的 Binary GCD 完全一样: 文献[7, §2]的关于 gcd 的几个定理(即算法的数学原理)正是文献[3, p. 321]的 four simple facts、文献[1, p. 66-67]的习题4 和文献[6, §4.2]的关于 gcd 的若干性质. 文献[7]虽引用了文献[1],但没注意到这个二进制 gcd 算法正是文献[1]的这个习题4.

文献[7]误认为二进制 gcd 算法在时间复杂度上比经典的 Euclid 算法有从 $O(\ln^3 N)$ 到 $O(\ln^2 N)$ 的数量级的改进,这是因为文献[7]不知道经典的 Euclid 算法的时间复杂度也为 $O(\ln^2 N)$ 这个事实. 经典的 Euclid 算法和二进制 gcd 算法耗时都是 $O(\ln^2 N)$,区别就在于大 O 中的常数了. 如果仔细编程(例如用汇编语言编程)使奇偶校验和二进制位移运算耗时占总耗时比例很小,二进制 gcd 算法就会比经典的 Euclid 算法稍快(参见文献[4]第15页的注解(2)和该页的最后一段).

* **收稿日期**:2002-12-10;**修改日期**:2003-11-25
作者简介:孙翠芳,女,1978 年生,助教,硕士生. 研究方向: 计算数论及其应用. E-mail: nick325@sohu.com

文献[7, §4]在谈到算法实现时取 $2^{16}=65\ 536$ 为多精度的基,但没提及所用的计算机机型. 这个算法之所以被称为"二进制 gcd 算法",就意味着要求程序设计者用汇编语言编程并取多精度的基为自己所用计算机的字长(2 的某个方幂). 现在 32 位机(PC386 以上)早已普及,在这样的机器上仍取多精度的基为 2^{16} 就会使机器性能闲置. 我们 NSFC 课题组在 PC Pentium Ⅲ/800 上 1 600 h 的计算工作[8]就是使用张振祥研制的基为 $2^{32}=65536^2=4\ 294\ 967\ 296$ 的多精度软件包,比在同样的机器上取基为 65 536 时速度快一倍.

最后我们指出,文献[7, § 4]中的关于"当 a、b 相差较大时先做一次带余除法(文献[7]称为'模除')"的做法正是文献[4, p. 15]的注解(3).

参 考 文 献

[1] Rosen K H. Elementary Number Theory and Its Applications[M]. Massachusetts: Addison Wesley, Reading, 1984.

[2] 张振祥,裴定一. 多重精度算术的时间复杂度分析[J]. 数学的实践与认识, 1994, (3): 74-76.

[3] Knuth D E. The Art of Computer Programming: Semi-numerical Algorithms, Vol. 2, 2nd ed., [M]. Massachusetts: Addison Wesley, Reading, 1981.

[4] Cohen H. A Course in Computational Algebraic Number Theory, 3., corr. print, Graduate Texts in Mathematics 138 [M]. Berlin: Springer-Verlag, 1996.

[5] ZHANG Zhenxiang. Using Lucas sequences to factor large integers near group orders[J]. The Fibonacci Quarterly, 2001, 39(3): 228-237.

[6] 卢开澄. 计算机密码学,第二版[M]. 北京: 清华大学出版社, 1998.

[7] 罗永龙,黄刘生,周智. 一个快速的二进制多重精度 gcd 算法[J]. 中国科学技术大学学报, 2002, 32(5): 542-545.

[8] ZHANG Zhenxiang and TANG Min. Finding strong pseudoprimes to several bases. Ⅱ[J]. Mathematics of Computation, 2003, 72(244): 2 085-2 097.

Notes on the Binary GCD Algorithm

SUN Cui-fang

(*Department of Mathematics, Anhui Normal University, Wuhu* 241000, *China*)

Abstract: Luo et al wrote in a recent paper [A Fast Algorithm for Computing gcd Based on Binary Multi-Precision, this journal, 2002, Vol. 32, No. 5, pp. 542-545; MR 2003h:11161] that "the classical Euclid's algorithm for computing the gcd of two integers takes time $O(\ln^3 N)$", and "present" an improved algorithm (called "binary gcd" for short) based on binary multi-precision with time complexity $O(\ln^2 N)$. In this paper, we point out two well-known facts: firstly, the binary gcd, without usefull implimentation improvements, is identical in mathematical theory to Stein's Binary GCD algorithm published in 1967; secondly, both Euclid's algorithm and Binary GCD have the same time complexity $O(\ln^2 N)$.

Key words: greatest common divisors (gcd); Euclid's algorithm; binary gcd