Научна конференция „Иновационни ИКТ в научните изследвания и обучението: математика, информатика и информационни технологии“, 29-30 ноември 2018 г., Пампорово, България
Scientific Conference “Innovative ICT in Research and Education: Mathematics, Informatics and Information Technologies”, 29-30 November 2018, Pamporovo, Bulgaria

THE FASTER LEHMER'S GREATEST COMMON DIVISOR ALGORITHM

Viktor Matanski, Pavel Kyurkchiev

Abstract: Here we will give faster modification of Lehmer's optimized solution for computing greatest common divisor. Our result is a natural continuation of new realizations given in [12]–[24]. The approach [12] is a good base for optimized solution for greatest common divisor. We also cite many world famous sources that study in details this topic [3]–[11], [25]–[35].

Keywords: *Euclidean algorithm for greatest common divisor, Lehmer's algorithm for greatest common divisor, reduced number of operations, reduced number of iterations*

Introduction

Let x and y be positive integers in radix b representation. Without loss of generality, assume that x and y have the same number of base b digits throughout. Lehmer's algorithm for computing greatest common divisor is the following [35]:

Algorithm 1.

INPUT: two positive integers x and y in radix b representation, with $x \geq y$.

OUTPUT: $gcd(x, y)$.

1. While $y \geq b$ do the following:

1.1 Set $\tilde{x}, \tilde{y}$ to be the high-order digit of x, y respectively ($\tilde{y}$ could be 0).

1.2 $A \leftarrow 1, B \leftarrow 0, C \leftarrow 0, D \leftarrow 1.$

1.3 While $(\tilde{y}+C) \neq 0$ and $(\tilde{y}+D) \neq 0$ do the following:

$$q \leftarrow \lfloor (\tilde{x}+A)/(\tilde{y}+C) \rfloor,\ q' \leftarrow \lfloor (\tilde{x}+B)/(\tilde{y}+D) \rfloor.$$

Дата на получаване: 03.12.2018 г.
Дата на рецензиране: 29.03.2019 г.
Дата на публикуване: 17.05.2019 г.

If $q \neq q'$, then go to 1.4.

$t \leftarrow A - qC,\ A \leftarrow C,\ C \leftarrow t,\ t \leftarrow B - qD,\ B \leftarrow D,\ D \leftarrow t.$

$t \leftarrow \tilde{x} - q\tilde{y},\ \tilde{x} \leftarrow \tilde{y},\ \tilde{y} \leftarrow t.$

1.4 If B = 0, then $T \leftarrow x \bmod y,\ x \leftarrow y,\ y \leftarrow T$;

otherwise, $T \leftarrow Ax + By,\ u \leftarrow Cx + Dy,\ x \leftarrow T,\ y \leftarrow u.$

2. While $y \neq 0$ do the following:

2.1 Set $r \leftarrow x \bmod y,\ x \leftarrow y,\ y \leftarrow r.$

3. Return(x).

We will point out that there are several implementation notes of Algorithm 1. which are presented in [35]:

- T is a multiple-precision variable. A, B, C, D, and t are signed single-precision variables. One bit of each of these variables must be reserved for the sign.
- The first operation of step 1.3 may result in overflow since $0 \leq \tilde{x} + A,\ \tilde{y} + D \leq b$. One solution is to reserve two bits more than the number of bits in a digit for each of x and y to accommodate both the sign and the possible overflow.
- The multiple-precision additions of step 1.4 are actually subtractions, since $AB \leq 0$ and $CD \leq 0$.

Main results

Using results given in [12] we receive the following optimized Lehmer's greatest common divisor algorithm:

Algorithm 2.

INPUT: two positive integers x and y in radix b representation.

OUTPUT: $gcd(x, y)$.

0. If $x > y$, then

1. While $y \geq b$ do the following:

1.1 Set $\tilde{x}, \tilde{y}$ to be the high-order digit of x, y respectively ($\tilde{y}$ could be 0).

1.2 $A \leftarrow 1,\ B \leftarrow 0,\ C \leftarrow 0,\ D \leftarrow 1.$

1.3 While $(\tilde{y} + C) \neq 0$ and $(\tilde{y} + D) \neq 0$ do the following:

$$q \leftarrow \lfloor (\tilde{x} + A)/(\tilde{y} + C) \rfloor,\ q' \leftarrow \lfloor (\tilde{x} + B)/(\tilde{y} + D) \rfloor.$$

If $q \neq q'$, then go to 1.4.

$t \leftarrow A - qC,\ A \leftarrow C,\ C \leftarrow t,\ t \leftarrow B - qD,\ B \leftarrow D,\ D \leftarrow t.$

$t \leftarrow \tilde{x} - q\tilde{y},\ \tilde{x} \leftarrow \tilde{y},\ \tilde{y} \leftarrow t.$

1.4 If B = 0, then $T \leftarrow x \bmod y,\ x \leftarrow y,\ y \leftarrow T;$

otherwise, $T \leftarrow Ax + By,\ u \leftarrow Cx + Dy,\ x \leftarrow T,\ y \leftarrow u.$

2. Do the following:

Set $x \leftarrow x \bmod y$,

If $x < 1$ Return(y), break.

Set $y \leftarrow y \bmod x$,

If $y < 1$ Return(x), break.

While true.

3. else of step 0. {i.e. the case $x \le y$}:

4. While $x \ge b$ do the following:

4.1 Set $\tilde{y}, \tilde{x}$ to be the high-order digit of y, x, respectively ($\tilde{x}$ could be 0).

4.2 $A \leftarrow 1,\ B \leftarrow 0,\ C \leftarrow 0,\ D \leftarrow 1.$

4.3 While $(\tilde{x} + C) \neq 0$ and $(\tilde{x} + D) \neq 0$ do the following:

$$q \leftarrow \lfloor (\tilde{y} + A)/(\tilde{x} + C) \rfloor,\ q' \leftarrow \lfloor (\tilde{y} + B)/(\tilde{x} + D) \rfloor.$$

If $q \neq q'$, then go to 4.4.

$t \leftarrow A - qC,\ A \leftarrow C,\ C \leftarrow t,\ t \leftarrow B - qD,\ B \leftarrow D,\ D \leftarrow t.$

$t \leftarrow \tilde{y} - q\tilde{x},\ \tilde{y} \leftarrow \tilde{x},\ \tilde{x} \leftarrow t.$

4.4 If B = 0, then $T \leftarrow y \bmod x,\ y \leftarrow x,\ x \leftarrow T;$

otherwise, $T \leftarrow Ay + Bx,\ u \leftarrow Cy + Dx,\ y \leftarrow T,\ x \leftarrow u.$

5. Do the following

Set $y \leftarrow y \bmod x$.

If $y < 1$ Return(x), break.

Set $x \leftarrow x \bmod y$.

If $x < 1$ Return(y), break.

While true.

Implementation notes of Algorithm 1. remain the same for Algorithm 2. and we will add the following to them:

- The first operation of step 4.3 may result in overflow since $0 \le \tilde{y} + A$, $\tilde{x} + D \le b$. One solution is to reserve two bits more than the number of bits in a digit for each of y and x to accommodate both the sign and the possible overflow.

- The multiple-precision additions of step 4.4 are actually subtractions, since AB ≤ 0 and CD ≤ 0.

Algorithm 2. uses algorithmic technique "divide and conquer" [12] where two basic branches are $x > y$ and $x \leq y$.

Acknowledgments

This work has been supported by the project FP17-FMI008 of Department for Scientific Research, University of Plovdiv "Paisii Hilendarski".

References

[1] A. Akritas, A new method for computing polynomial greatest common divisors and polynomial remainder sequences, *Numerische Mathematik*, 52 (1988), 119–127.

[2] A. Akritas, G. Malaschonok, P. Vigklas, On the Remainders Obtained in Finding the Greatest Common Divisor of Two Polynomials, *Serdica Journal of Computing*, 9 (2015), 123–138.

[3] L. Ammeraal, *Algorithms and Data Structures in C++*, John Wiley & Sons Inc., New York (1996).

[4] D. Bressoud, *Factorization and primality testing*, Springer-Verlag, New York (1989).

[5] F. Chang, Factoring a Polynomial with Multiple-Roots, *World Academy of Science, Engineering and Technology*, 47 (2008), 492–495.

[6] Th. Cormen, Ch. Leiserson, R. Rivest, Cl. Stein, *Introduction to Algorithms*, 3rd ed., The MIT Press, Cambridge (2009).

[7] A. Drozdek, *Data Structures and Algorithms in C++*, 4th ed., Cengage Learning (2013).

[8] K. Garov, A. Rahnev, *Textbook-notes on programming in BASIC for facultative training in mathematics for 9.–10. Grade of ESPU*, Sofia (1986). (in Bulgarian)

[9] S. Goldman, K. Goldman, *A Practical Guide to Data Structures and Algorithms Using JAVA*, Chapman & Hall/CRC, Taylor & Francis Group, New York (2008).

[10] A. Golev, *Textbook on algorithms and programs in C#*, University Press "Paisii Hilendarski", Plovdiv (2012).

[11] M. Goodrich, R. Tamassia, D. Mount, *Data Structures and Algorithms in C++*, 2nd ed., John Wiley & Sons Inc., New York (2011).

[12] A. Iliev, N. Kyurkchiev, A Note on Knuth's Implementation of Euclid's Greatest Common Divisor Algorithm, *International Journal of Pure and Applied Mathematics*, 117 (2017), 603–608.

[13] A. Iliev, N. Kyurkchiev, A. Golev, A Note on Knuth's Implementation of Extended Euclidean Greatest Common Divisor Algorithm, *International Journal of Pure and Applied Mathematics*, 118 (2018), 31–37.

[14] A. Iliev, N. Kyurkchiev, A. Rahnev, A Note on Adaptation of the Knuth's Extended Euclidean Algorithm for Computing Multiplicative Inverse, *International Journal of Pure and Applied Mathematics*, 118 (2018), 281–290.

[15] A. Iliev, N. Kyurkchiev, A Note on Euclidean and Extended Euclidean Algorithms for Greatest Common Divisor for Polynomials, *International Journal of Pure and Applied Mathematics*, 118 (2018), 713–721.

[16] A. Iliev, N. Kyurkchiev, A Note on Least Absolute Remainder Euclidean Algorithm for Greatest Common Divisor, *International Journal of Scientific Engineering and Applied Science*, 4(3) (2018), 31–34.

[17] A. Iliev, N. Kyurkchiev, A Note on Knuth's Algorithm for Computing Extended Greatest Common Divisor using SGN Function, *International Journal of Scientific Engineering and Applied Science*, 4(3) (2018), 26–29.

[18] A. Iliev, N. Kyurkchiev, *New Trends in Practical Algorithms: Some Computational and Approximation Aspects*, LAP LAMBERT Academic Publishing, Beau Bassin (2018).

[19] A. Iliev, N. Kyurkchiev, 80th Anniversary of the birth of Prof. Donald Knuth, *Biomath Communications*, 5 (2018), 7 pp.

[20] A. Iliev, N. Kyurkchiev, New Realization of the Euclidean Algorithm, *Collection of scientific works of Eleventh National Conference with International Participation Education and Research in the Information Society*, Plovdiv, ADIS, June 1–2, (2018), 180–185. (in Bulgarian)

[21] A. Iliev, N. Kyurkchiev, New Organizing of the Euclid's Algorithm and one of its Applications to the Continued Fractions, *Collection of scientific works from conference "Mathematics. Informatics. Information Technologies. Application in Education"*, Pamporovo, Bulgaria, October 10–12, (2018). (to appear)

[22] A. Iliev, N. Kyurkchiev, The faster Euclidean algorithm, Proc. of *Scientific Conference "Innovative ICT in Research and Education: Mathematics, Informatics and Information Technologies"*, Pamporovo, Bulgaria, November 28–30, 15–20, 2018.

[23] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, Proc. of *Scientific Conference "Innovative ICT in Research and Education: Mathematics, Informatics and Information Technologies"*, Pamporovo, Bulgaria, November 28–30, 21–26, 2018.

[24] P. Kyurkchiev, V. Matanski, The faster Euclidean algorithm for computing polynomial multiplicative inverse, Proc. of *Scientific Conference "Innovative ICT in Research and Education: Mathematics, Informatics and Information Technologies"*, Pamporovo, Bulgaria, November 29–30, 21–26, 2018.

[25] A. Iliev, N. Valchanov, T. Terzieva, Generalization and Optimization of Some Algorithms, *Collection of scientific works of National Conference "Education in Information Society"*, Plovdiv, ADIS, May 12–13, (2009), 52–58, (in Bulgarian).
[26] D. Knuth, *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, 3rd ed., Addison-Wesley, Boston (1998).
[27] Hr. Krushkov, *Programming in C#*, Koala press, Plovdiv (2017). (in Bulgarian)
[28] P. Nakov, P. Dobrikov, *Programming=++Algorithms*, 5th ed., Sofia (2015). (in Bulgarian)
[29] A. Rahnev, K. Garov, O. Gavrailov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia (1985). (in Bulgarian)
[30] A. Rahnev, K. Garov, O. Gavrailov, *BASIC in examples and tasks*, Government Press "Narodna prosveta", Sofia (1990). (in Bulgarian)
[31] D. Schmidt, *Euclid's GCD Algorithm* (2014).
[32] R. Sedgewick, K. Wayne, *Algorithms*, 4th ed., Addison-Wesley, Boston (2011).
[33] A. Stepanov, *Notes on Programming* (2007).
[34] R. Crandall, C. Pomerance, *Prime Numbers: A Computational Perspective*, Springer-Verlag, New York (2005).
[35] A. Menezes, P. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, 5th ed., CRC Press LLC, New York (2001).

Faculty of Mathematics and Informatics,
University of Plovdiv Paisii Hilendarski,
24, Tzar Asen Str., 4000 Plovdiv, Bulgaria,
e-mails: viktor_matanski@yahoo.com, pkyurkchiev@uni-plovdiv.bg

ПО-БЪРЗИЯТ АЛГОРИТЪМ НА ЛЕМЕР ЗА НАЙ-ГОЛЕМИЯ ОБЩ ДЕЛИТЕЛ

Виктор Матански, Павел Кюркчиев

Резюме: Тук даваме по-бърза модификация на оптимизираното решение от Лемер за изчисляване на най-големия общ делител. Нашият резултат е естествено продължение на новите реализации дадени в [12]–[24]. Подходът [12] е добра основа за оптимизирано решение за най-големия общ делител. Цитираме също много световноизвестни източници, които детайлно изучават тази тематика [3]–[11], [25]–[35].