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 EXTENDED EUCLIDEAN ALGORITHM

Anton Iliev, Nikolay Kyurkchiev

Abstract: In our previous works [12]–[22] we give optimized solutions of Euclidean algorithm. Computational effectiveness of these algorithms [12]–[22] make them more useful from practical point of view in comparison to [3]–[11], [23]–[31]. Using the known general analysis of extended Euclid's algorithm we give theorem which approve correctness for new [12]–[22] suggested by us extended Euclidean algorithm which is one of the most used.

Keywords: *greatest common divisor, extended Euclid's algorithm, reduced memory usage*

1. Introduction

Let $a > 0$ and $b > 0$ be a natural numbers and by $m (m \geq 1)$ we denote the number of divisions in the extended Euclid's algorithm. Without loosing of generality we will explore the case when $a > b$.

2. Main results

Theorem. Let a and b be a natural numbers and let their greatest common divisor is denoted by $g = gcd(a, b)$. Then there are integers x and y for that $xa + yb = g$.

Proof. We use the Euclid's algorithm [22] for $a_0 = a$ and $a_1 = b$ (analysis of the other case $a \leq b$ is analogical because the only that we need is to swap a and b). If m is even number the iteration procedure [12]–[22] can be expressed by the following:

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

$$
\begin{aligned}
&\textit{First step}:\\
&\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} c_0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} c_1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 \\ a_3 \end{pmatrix};\\
&\textit{Second step}:\\
&\begin{pmatrix} a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} c_2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_3 \\ a_4 \end{pmatrix}, \begin{pmatrix} a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} c_3 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_4 \\ a_5 \end{pmatrix};\\
&\ldots\\
&\lfloor (m+1)/2 \rfloor\, \textit{step}:\\
&\begin{pmatrix} a_{m-2} \\ a_{m-1} \end{pmatrix} = \begin{pmatrix} c_{m-2} & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{m-1} \\ a_m \end{pmatrix}, \begin{pmatrix} a_{m-1} \\ a_m \end{pmatrix} = \begin{pmatrix} c_{m-1} & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_m \\ a_{m+1} \end{pmatrix},
\end{aligned}
\tag{1}
$$

where $c_j = \lfloor a_j / a_{j+1} \rfloor,\ 0 \le j \le m-1$.

As we already mention in [22] – it is obvious that if m is odd number then every "step" will take one-half of "some step" of (1) and eventually one-half from "next step" of (1) which is only technical aspect in computations' organizing. This division in "steps" is only for easier explanation of general idea.

We will explore the case when m is an even number (the case when m is an odd number is analogical as we noted in [22]).

So, from (1) we receive that $g = a_m = gcd(a_0, a_1),\ a_{m+1} = 0$ and $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c_0 & 1 \\ 1 & 0 \end{pmatrix} \ldots \begin{pmatrix} c_{m-1} & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} g \\ 0 \end{pmatrix}$.

We set $N_s = \begin{pmatrix} d_s & e_s \\ f_s & g_s \end{pmatrix} = \begin{pmatrix} c_0 & 1 \\ 1 & 0 \end{pmatrix} \ldots \begin{pmatrix} c_s & 1 \\ 1 & 0 \end{pmatrix}$ and consequently $N_{m-1}^{-1}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$.

Because $\det N_{m-1} = (-1)^m$ we obtain $N_{m-1}^{-1} = (-1)^m \begin{pmatrix} g_{m-1} & -e_{m-1} \\ -f_{m-1} & d_{m-1} \end{pmatrix}$.

From here $g_{m-1}a - e_{m-1}b = (-1)^m g$ and $x = (-1)^m g_{m-1},\ y = (-1)^{m+1} e_{m-1}$.

This process can be written in this algorithmic form:

```
Euclid(a, b, ref x, ref y)
       a2 = a mod b
```

```
        c0 = a / b
        if (a2 < 1)
                x = 1
                y = 0
                return b
        a3 = b mod a2
        c1 = b / a2
        if (a3 < 1)
                x = – c0
                y = 1
                return a2
        g = Euclid(a2, a3, ref x, ref y)
        y – = c1*x; x – = c0*y
        return g
  and the calling is:
     if (a > b)
           Euclid(a, b, ref y, ref x)
         else
           Euclid(b, a, ref x, ref y)
```

Acknowledgments

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

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] Chang, F., 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, *Proc. 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, *Proc. of Scientific 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 29–30, 15–20, 2018.

[23] A. Iliev, N. Valchanov, T. Terzieva, Generalization and Optimization of Some Algorithms, *Proc. of National Conference "Education in Information Society"*, Plovdiv, ADIS, May 12–13, (2009), 52–58, (in Bulgarian).

[24] D. Knuth, *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, 3rd ed., Addison-Wesley, Boston (1998).

[25] Hr. Krushkov, *Programming in C#*, Koala press, Plovdiv (2017). (in Bulgarian)

[26] P. Nakov, P. Dobrikov, *Programming=++Algorithms*, 5th ed., Sofia (2015). (in Bulgarian)

[27] A. Rahnev, K. Garov, O. Gavrailov, *Textbook for extracurricular work using BASIC*, MNP Press, Sofia (1985). (in Bulgarian)

[28] A. Rahnev, K. Garov, O. Gavrailov, *BASIC in examples and tasks*, Government Press "Narodna prosveta", Sofia (1990). (in Bulgarian)

[29] D. Schmidt, *Euclid's GCD Algorithm* (2014).

[30] R. Sedgewick, K. Wayne, *Algorithms*, 4th ed., Addison-Wesley, Boston (2011).

[31] A. Stepanov, *Notes on Programming* (2007).

Faculty of Mathematics and Informatics,

University of Plovdiv "Paisii Hilendarski",

24 Tzar Asen Str., 4000 Plovdiv, Bulgaria,

e-mails: aii@uni-plovdiv.bg, nkyurk@uni-plovdiv.bg

ПО-БЪРЗИЯТ РАЗШИРЕН АЛГОРИТЪМ НА ЕВКЛИД

Антон Илиев, Николай Кюркчиев

Резюме: В наши предишни работи [12]–[22] даваме оптимизирани решения за алгоритъма на Евклид. Изчислителната ефективност на тези алгоритми [12]–[22] ги прави по-полезни от практическа гледна точка в сравнение с [3]–[11], [23]–[31]. Използвайки известния общ анализ на разширения алгоритъм на Евклид, даваме теорема, която утвърждава коректността на новия [12]–[22] предложен от нас разширен алгоритъм на Евклид, който е един от най-използваните.