Neural, Parallel, and Scientific Computations, **27**, No. 1 (2019), 1-9 ISSN: 1061-5369

A NEW IMPROVEMENT OF LEAST ABSOLUTE REMAINDER ALGORITHM FOR GREATEST COMMON DIVISOR. III

ANTON ILIEV[1], NIKOLAY KYURKCHIEV[2], AND ASEN RAHNEV[3]
[1,2,3]Faculty of Mathematics and Informatics
University of Plovdiv Paisii Hilendarski
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA

ABSTRACT: In this note we gave new highly optimized realization of least absolute remainder algorithm for calculation of greatest common divisor (GCD). Here we give new organizing way which is different from presented in [1]–[26], [43]–[66]. For computer implementation Visual C# 2017 programming environment is used.

AMS Subject Classification: 11A05, 68W01
Key Words: greatest common divisor, Euclid's algorithm, improvement algorithm, least absolute remainder, reduced number of iterations

Received: October 12, 2018 ; **Accepted:** December 10, 2018 ;
Published: December 11, 2019. **doi:** 10.12732/npsc.v27i1.1
Dynamic Publishers, Inc., Acad. Publishers, Ltd. https://acadsol.eu/npsc

1. INTRODUCTION

For previous results see [27]–[42]. Here we are concentrated to receive faster solution for GCD.

2. MAIN RESULTS

Now we set the task to find more effective Euclidean GCD algorithm. For testing we will use the following computer: processor - Intel(R) Core(TM) i7-6700HQ CPU 2.60GHz, 2592 Mhz, 4 Core(s), 8 Logical Processor(s), RAM 16 GB, Microsoft Windows 10 Enterprise x64.

Let $a>0$ and $b>0$ be natural numbers. Least absolute remainder algorithm [66] is known:

Algorithm 1.

```
if (a > b) do { r = a %= b;
if (a == 0) { gcd = b; break; }
ar = b - a; a = b;
if (r <= ar) b = r; else b = ar;
} while (true);
else do { r = b %= a;
if (b == 0) { gcd = a; break; }
ar = a - b; b = a;
if (r <= ar) a = r; else a = ar;
} while (true);
```

Its recursive implementation is:

Algorithm 2.

```
static long Euclid(long a, long b)
{ long r = a %= b;
if (a == 0) return b;
long ar = b - a;
a = b;
if (r <= ar) b = r; else b = ar;
return Euclid(a, b); }
```

We present the optimized iterative realization of Algorithm 1

Algorithm 3.

```
if (a > b) do { a %= b;
if (a == 0) { gcd = b; break; }
ar = b - a;
if (a > ar) a = ar;
b %= a;
if (b == 0) { gcd = a; break; }
ar = a - b;
if (b > ar) b = ar;
} while (true);
else do { b %= a;
if (b == 0) { gcd = a; break; }
ar = a - b;
if (b > ar) b = ar;
a %= b;
if (a == 0) { gcd = b; break; }
```

```
ar = b - a;
if (a > ar) a = ar;
} while (true);
```

and optimized recursive realization of Algorithm 2

Algorithm 4.

```
static long Euclid(long a, long b)
{ a %= b;
if (a == 0) return b;
long ar = b - a;
if (a > ar) a = ar;
b %= a;
if (b == 0) return a;
ar = a - b;
if (b > ar) b = ar;
return Euclid(a, b); }
```

Numerical experiment:

Part 1.

```
long a, b, r, ar, gcd, d = 0;
for (int i = 1; i < 100000001; i++) { b = i; a = 200000002 - i;
//here is the source code of every one of algorithms 1, 3
//and calling of recursive algorithms 2 and 4
d += gcd; }
Console.WriteLine(d);
```

Part 2. We will use the task from Part 1. where we swapped the values of 'a' and 'b'.

Part 3. Average time of performance

EN = (Part 1.Algorithm$N$ + Part 2.Algorithm$N$) / 2,
where $N = 1$ to 4 denotes using of Algorithms 1 to 4.

Both recursive implementations can be called by:

```
if (a > b) gcd = Euclid(a, b); else gcd = Euclid(b, a);
```

We will point out that solutions (see Fig. 1 and Fig. 2) presented here (recursive - Algorithm 4 and iterative Algorithm 3) in computational aspect are more effective even than these in [41].

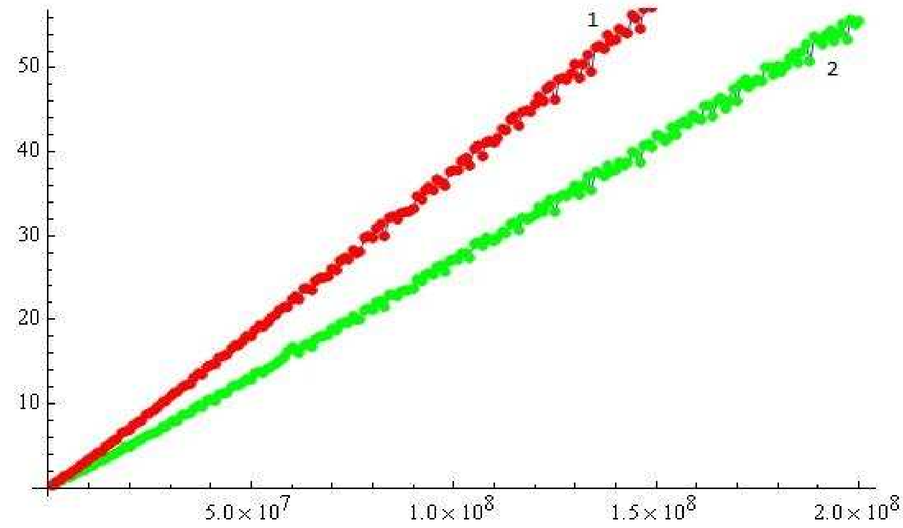

Figure 1: Algorithm 2 - Least absolute remainder recursive (1 – red color), Algorithm 4 - Iliev–Kyurkchiev–Rahnev recursive (2 – green line)

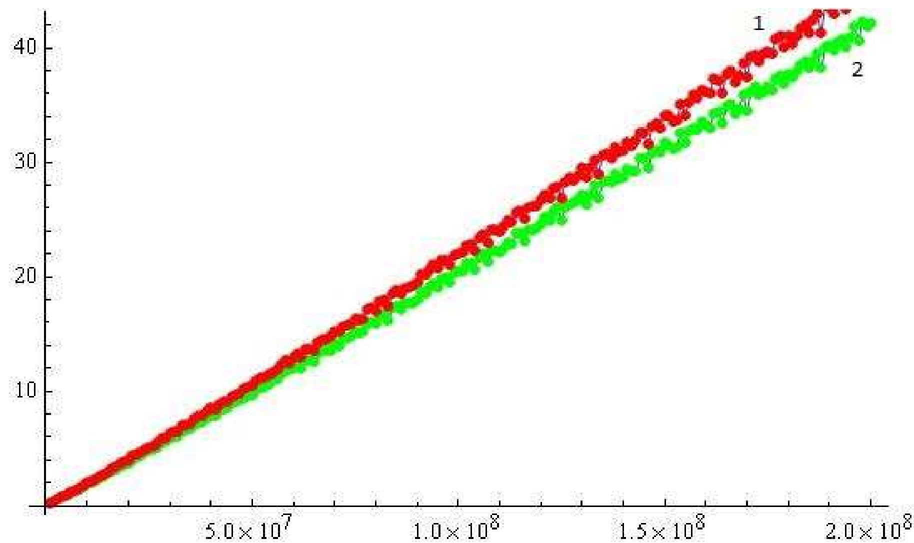

Figure 2: Algorithm 1 - Least absolute remainder iterative (1 – red color), Algorithm 3 - Iliev–Kyurkchiev–Rahnev iterative (2 – green line)

ACKNOWLEDGMENTS

This work has been supported by the project BG05M2OP001-1.001-0003-01 “CoE on Informatics and ICT” supported by the Operational Programme “Science and Education for Smart Growth”.

REFERENCES

[1] A. Aho, J. Hopcroft, J. Ullman, *The Design and Analysis of Computer Algorithms*, 1st ed., Addison-Wesley, Boston (1974).

[2] A. Aho, J. Ullman, J. Hopcroft, *Data Structures and Algorithms*, 1st ed., Addison-Wesley, Boston (1987).

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

[4] 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.

[5] M. Alsuwaiyel, *Algorithms: Design Techniques and Analysis*, Lecture Notes Series on Computing, revised ed., World Scientific Publishing Company, Hackensack (2016).

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

[7] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer-Verlag, New York (1976).

[8] S. Baase, A. Gelder, *Computer Algorithms, Introduction to Design and Analysis*, 3rd ed., Addison-Wesley, Boston (2000).

[9] G. Brassard, P. Bratley, *Fundamentals of Algorithmics*, international ed., Pearson, (2015).

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

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

[12] Th. Cormen, *Algorithms Unlocked*, MIT Press, Cambridge (2013).

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

[14] R. Crandall, C. Pomerance, *Prime Numbers: A Computational Perspective*, Springer-Verlag, New York (2005).

[15] J. D. Dixon, The number of steps in the Euclidean algorithm, *J. Number Theory*, **2** (1970), 414–422.

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

[17] J. Erickson, *Algorithms*, University of Illinois Press (2009).

[18] J. Gareth, J. Jones, *Elementary Number Theory*, Springer-Verlag, New York (1998).

[19] 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)

[20] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer, New York (1996).

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

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

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

[24] R. Graham, D. Knuth, O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed., Addison-Wesley, Boston (1994).

[25] D. H. Greene, D. E. Knuth, *Mathematics for the Analysis of Algorithms*, 2nd ed., Birkhauser, Boston (1982).

[26] H. A. Heilbronn, On the average length of a class of finite continued fractions. In: *Number Theory and Analysis (Turan, P., ed.)*, 87–96, Plenum Press, New York (1969).

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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**, No. 3 (2018), 31–34.

[32] 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**, No. 3 (2018), 26–29.

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

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

[35] 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)

[36] 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)

[37] A. Iliev, N. Kyurkchiev, The faster Euclidean algorithm, *Collection of scientific works from conference, Pamporovo, Bulgaria*, November 28–30, (2018). (to appear)

[38] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, *Collection of scientific works from conference, Pamporovo, Bulgaria*, November 28–30, (2018). (to appear)

[39] P. Kyurkchiev, V. Matanski, The faster Euclidean algorithm for computing polynomial multiplicative inverse, *Collection of scientific works from conference, Pamporovo, Bulgaria*, November 28–30, (2018). (to appear)

[40] V. Matanski, P. Kyurkchiev, The faster Lehmer's greatest common divisor algorithm, *Collection of scientific works from conference, Pamporovo, Bulgaria*, November 28–30, (2018). (to appear)

[41] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement Euclidean Algorithm for Greatest Common Divisor. I, *Neural, Parallel, and Scientific Computations*, **26**, No. 3 (2018), 355–362.

[42] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Harris-Stein Modification of Euclidean Algorithm for Greatest Common Divisor. II, *International Journal of Pure and Applied Mathematics*, **120**, No. 4 (2018). (to appear)

[43] 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), http://sci-gems.math.bas.bg/jspui/handle/10525/1356

[44] E. Kaltofen, H. Rolletschek, Computing greatest common divisors and factorizations in quadratic number fields, *Math. Comp.*, **53** (1990), 697–720.

[45] J. Kleinberg, E. Tardos, *Algorithm Design*, Addison-Wesley, Boston (2006).

[46] D. E. Knuth, Evaluation of Porter's constant, *Comp. Maths. Appls.*, **2** (1976), 137–139.

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

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

[49] A. Levitin, *Introduction to the Design and Analysis of Algorithms*, 3rd ed., Pearson, Boston (2011).

[50] A. Menezes, P. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, 5th ed., CRC Press LLC, New York (2001).

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

[52] G. H. Norton, A shift-remainder GCD algorithm. In: Applied Algebra. Algebraic Algorithms and Error Correcting Codes (Huguet, L., Poli, A., eds.), *Springer LNCS*, **356** (1989), 350–356.

[53] G. H. Norton, On the Asymptotic Analysis of the Euclidean Algorithm, *J. Symbolic Computation*, **10** (1990), 53–58.

[54] J. W. Porter, On a theorem of Heilbronn, *Mathematika*, **22** (1975), 20–28.

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

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

[57] H. Rolletschek, On the number of divisions of the Euclidean algorithm applied to Gaussian integers, *J. Symbolic Computation*, **2** (1986), 261–291.

[58] H. Rolletschek, Shortest division chains in imaginary quadratic number fields. In: Symbolic and Algebraic Computation (Gianni, P., ed.), *Springer LNCS* **358** (1990), 231–243.

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

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

[61] S. Skiena, *The Algorithm Design Manual*, 2nd ed., Springer, New York (2008).

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

[63] E. Weisstein, *CRC Concise Encyclopedia of Mathematics*, Chapman & Hall/CRC, New York (2003).

[64] J. Stein, Computational problems associated with Racah algebra, *Journal of Computational Physics*, **1** (1967), 397–405.

[65] V. Harris, An algorithm for finding the greatest common divisor, *Fibonacci Quarterly*, **8** (1970), 102–103.

[66] T. Moore, On the Least Absolute Remainder Euclidean Algorithm, *Fibonacci Quarterly*, **30** (1992), 161–165.