International Electronic Journal of Pure and Applied Mathematics
Volume 14 No. 1 2020, 37-44
ISSN: 1314-0744
url: https://www.e.ijpam.eu
doi: 10.12732/iejpam.v14i1.5

A NEW IMPROVEMENT OF MODULAR MULTIPLICATIVE INVERSE BINARY ALGORITHM

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 paper we present binary algorithm for finding modular multiplicative inverse. We continue our research of development optimal or near to optimal algorithms [7]–[33] for the class of problems concern greatest common divisor. The algorithms provided by us demonstrate strong symmetry.

AMS Subject Classification: 11A05, 68W01
Key Words: modular multiplicative inverse, recursive binary algorithm, iterative binary algorithm, reduced number of operations

1. Introduction

Our new algorithms work correctly for every $a > 0$ and $b > 0$. They both iterative and recursive compute the multiplicative inverse of *a modulo b*; $a^{-1} \pmod{b}$, and returns either the inverse as a positive integer less than b, or 0 if no inverse exists. The multiplicative inverse of *a modulo b* exists if and only if a and b are coprime (i.e., if their greatest common divisor (gcd) is equal to 1). If the multiplicative inverse of *a modulo b* exists, the operation of division by *a modulo b* can be defined as multiplying by the inverse. We use also optimization which is that when both a and b are even we can skip computational process because $gcd >= 2$ and no modular multiplicative inverse exists. We demonstrate the approach which is different in comparison to computational way which is given in [1]–[6] and [33]–[41].

Received: May 29, 2020
Revised: October 19, 2020
Published: October 23, 2020

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 14, No. 1 (2020)

For testing purposes for new algorithm 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, Microsoft Visual C# 2017 x64.

2. Main Results

We propose the following iterative binary algorithm for finding modular multiplicative inverse –

Algorithm 1.

```
if ((a & 1) == 0 && (b & 1) == 0) { eeacmi = 0; }
else { x1 = 1; x2 = 0; y1 = 0; y2 = 1;
u = a; v = b;
while ((u & 1) == 0)
{ u >>= 1;
if ((x1 & 1) == 0 && (x2 & 1) == 0) { x1 >>= 1; x2 >>= 1; }
else { x1 = (x1 + b) >> 1; x2 = (x2 - a) >> 1; } }
while ((v & 1) == 0)
{ v >>= 1;
if ((y1 & 1) == 0 && (y2 & 1) == 0) { y1 >>= 1; y2 >>= 1; }
else { y1 = (y1 + b) >> 1; y2 = (y2 - a) >> 1; } }
while (u != v)
if (u > v) { u -= v; x1 -= y1; x2 -= y2;
while ((u & 1) == 0) { u >>= 1;
if ((x1 & 1) == 0 && (x2 & 1) == 0) { x1 >>= 1; x2 >>= 1; }
else { x1 = (x1 + b) >> 1; x2 = (x2 - a) >> 1; } } }
else { v -= u; y1 -= x1; y2 -= x2;
while ((v & 1) == 0) { v >>= 1;
if ((y1 & 1) == 0 && (y2 & 1) == 0) { y1 >>= 1; y2 >>= 1; }
else { y1 = (y1 + b) >> 1; y2 = (y2 - a) >> 1; } } }
if (v > 1) eeacmi = 0;
else { while (y1 < 0) y1 += b; while (y1 >= b) y1 -= b; eeacmi = y1; } }
```

and its recursive variant –

Algorithm 2.

```
static long Euclid(long a0, long b0, long a, long b, ref long x,
ref long y, long x1, long x2, long y1, long y2)
{ if ((a & 1) == 0)
```

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 14, No. 1 (2020)

```
{ if ((b & 1) == 0) return Euclid(a0 >> 1, b0 >> 1, a >> 1, b >> 1,
ref x, ref y, x1, x2, y1, y2) << 1;
else { if ((x1 & 1) == 0 && (x2 & 1) == 0)
{ x1 >>= 1; x2 >>= 1; }
else { x1 = (x1 + b0) >> 1; x2 = (x2 - a0) >> 1; }
return Euclid(a0, b0, a >> 1, b, ref x, ref y, x1, x2, y1, y2); } }
else if ((b & 1) == 0)
{ if ((y1 & 1) == 0 && (y2 & 1) == 0)
{ y1 >>= 1; y2 >>= 1; }
else { y1 = (y1 + b0) >> 1; y2 = (y2 - a0) >> 1; }
return Euclid(a0, b0, a, b >> 1, ref x, ref y, x1, x2, y1, y2); }
else if (a == b) { x = y1; y = y2; return a; }
else if (a > b)
return Euclid(a0, b0, a - b, b, ref x, ref y, x1 - y1, x2 - y2, y1, y2);
else
return Euclid(a0, b0, a, b - a, ref x, ref y, x1, x2, y1 - x1, y2 - x2); }
```

It can be called by:

```
if ((a & 1) == 0 && (b & 1) == 0) { eeacmi = 0; }
else { a0 = a; b0 = b;
x1 = 1; x2 = 0; y1 = 0; y2 = 1;
gcd = Euclid(a0, b0, a, b, ref x, ref y, x1, x2, y1, y2);
if (gcd > 1) eeacmi = 0;
else { while (x < 0) x += b; while (x >= b) x -= b; eeacmi = x; } }
```

Remark. Here we will present also Algorithm 3. for finding modular multiplicative inverse using in its processing part only "difference" operation.

Algorithm 3.

```
b0 = b;
if ((a & 1) == 0 && (b & 1) == 0) { eeacmi = 0; }
else
{ x1 = 1; x2 = 0; y1 = 0; y2 = 1;
while (a != b)
if (a > b) { a -= b; x1 -= y1; x2 -= y2; }
else { b -= a; y1 -= x1; y2 -= x2; }
if (a == 1) { if (y1 < 0) y1 += b0;
eeacmi = y1; }
else eeacmi = 0;
}
```

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 14, No. 1 (2020)

3. Numerical Example

We will test the proposed algorithms for the following example:

```
long a, b, x = 0, y = 0, y1, y2, d = 0, a0, b0;
long x1, x2, gcd, eeacmi, u, v;
for (int i = 1; i < 100000001; i++) { a = i; b = 200000002 - i;
//Here can be placed the source code of Algorithms 1 and 3
//and the calling of recursive variant of Algorithm 1  Algorithm 2.
d += eeacmi; }
Console.WriteLine(d);
```

CPU time of Algorithm 1 is: **34.923 seconds.**
CPU time of Algorithm 2 is: **106.229 seconds.**
CPU time of Algorithm 3 is: **55.402 seconds.**

4. Conclusion

We present new realizations of both – binary modular multiplicative inverse recursive and iterative algorithms respectively. The question of optimality for computational way and number of used operations is planned to be solved.

5. Acknowledgments

This paper is supported by the Project FP19-FMI-002 "Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education" of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

References

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

[2] S. Enkov, *Programming in Arduino Environment*, University Press "Paisii Hilendarski", Plovdiv (2017). (in Bulgarian)

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

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

[5] 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)

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

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

[15] 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)

[16] 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, 10–12 October 2018, (2019), 199–207.

[17] A. Iliev, N. Kyurkchiev, The faster Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 15–20.

[18] A. Iliev, N. Kyurkchiev, The faster extended Euclidean algorithm, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 21–26.

[19] P. Kyurkchiev, V. Matanski, The faster Euclidean algorithm for computing polynomial multiplicative inverse, *Collection of scientific works from conference*, Pamporovo, Bulgaria, 28–30 November 2018, (2019), 43–48.

[20] V. Matanski, P. Kyurkchiev, The faster Lehmer's greatest common divisor algorithm, *Collection of scientific works from conference, Pamporovo*, Bulgaria, 28–30 November 2018, (2019), 37–42.

[21] 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.

[22] 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. 3 (2018), 379–388.

[23] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Least Absolute Remainder Algorithm for Greatest Common Divisor. III, *Neural, Parallel, and Scientific Computations*, 27 No. 1 (2019), 1–9.

[24] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Tembhurne–Sathe Modification of Euclidean Algorithm for Greatest Common Divisor. IV, *Dynamic Systems and Applications*, 28 No. 1 (2019), 143–152.

[25] A. Iliev, N. Kyurkchiev, A. Rahnev, *Nontrivial Practical Algorithms: Part 2*, LAP LAMBERT Academic Publishing, Beau Bassin (2019).

[26] 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, 12–13 May 2009, (2009), 52–58. (in Bulgarian)

[27] H. Gyulyustan, A Note on Euclidean Sequencing Algorithm, *Proceedings of the Scientific Conference "Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education"*, Pamporovo, 7–8 November 2019, Plovdiv University Press, (2020), 57–64.

[28] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 69–74.

[29] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Stein's Binary Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 75–80.

[30] A. Iliev, N. Kyurkchiev, A. Rahnev, New Algorithms for Finding Modular Multiplicative Inverse, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 81–88.

[31] A. Iliev, N. Kyurkchiev, A. Rahnev, New Extended Algorithm for Finding Greatest Common Divisor, *Neural, Parallel, and Scientific Computations*, 28 No. 1 (2020), 89–95.

[32] A. Iliev, N. Kyurkchiev, A. Rahnev, Recursive Extended Stein's Binary Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 14 No. 1 (2020). (accepted)

[33] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Extended Stein's Binary Algorithm, *Proceedings of the Anniversary International Scientific Conference Synergetics and Reflection in Mathematics Education*, Pamporovo, 16–18 October 2020, Plovdiv University Press, (2020), 259–264.

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

[35] Hr. Krushkov, A. Iliev, *Practical programming guide in Pascal, Parts I and II*, Koala press, Plovdiv (2002). (in Bulgarian)

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

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

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

[39] N. Kasakliev, *C# Programming Guide*, University Press "Paisii Hilendarski", Plovdiv (2016). (in Bulgarian)

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 14, No. 1 (2020)

[40] A. Rahnev, N. Pavlov, N. Valchanov, T. Terzieva, *Object Oriented Programming*, Lightning Source UK Ltd., London (2014).

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

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 14, No. 1 (2020)