International Electronic Journal of Pure and Applied Mathematics
Volume 15 No. 1 2021, 1-11
ISSN: 1314-0744
url: https://www.e.ijpam.eu
doi: 10.12732/iejpam.v15i1.1

A NEW IMPROVEMENT OF JACOBI SYMBOL BINARY ALGORITHM

Anton Iliev[1], Nikolay Kyurkchiev[2], 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 research we present new iterative and recursive versions of Jacobi symbol binary algorithm. Our result speeding up the algorithm *BinaryJacobi* given in [46].

AMS Subject Classification: 11A05, 68W01
Key Words: Jacobi symbol, reduced number of operations

1. Introduction

For any integer number a and any natural odd number b, the binary algorithm for calculating Jacobi symbol is given in [46]. The theory of Euclidean algorithm is presented in many sources [1]–[46]. By innovative computational approach we show how these algorithms can be optimized [10]–[38] especially in recursive but as well as in iterative implementations.

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.

The binary calculation of Jacobi symbol is given in [46]:

Received: November 11, 2020
Revised: January 27, 2021
Published: February 9, 2021

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

Algorithm 1.

```
j = 1;
if (a < 0)
{
if ((b & 3) == 3) j = -j;
a = -a;
}
while (a > 0)
{
while ((a & 1) == 0)
{
a >>= 1; r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (a < b)
{
r = a; a = b; b = r;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
a = (a - b) >> 1;
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (b == 1) Jacobi = j; else Jacobi = 0;
```

and its recursive binary version is:

Algorithm 2.

```
static long Euclid(long a, long b, ref int j)
{
long r;
if (a == 0) return b;
else
{
if ((a & 1) == 0)
{
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid(a >> 1, b, ref j);
}
if (a < b)
```

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

```
{
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
return Euclid(b, a, ref j);
}
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid((a - b) >> 1, b, ref j);
}
}
```

Main Results

We will present new binary versions of Algorithms 1 and 2 as Algorithms 3, 5 and 4, 6 respectively which possess reduced number of operations.

The following line is included to the source code: if (a == 1) { b = 1; break; } which also optimizes Algorithm 1 and leads to Algorithm 3:

Algorithm 3.

```
j = 1;
if (a < 0)
{
if ((b & 3) == 3) j = -j;
a = -a;
}
while (a > 0)
{
while ((a & 1) == 0)
{
a >>= 1; r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (a == 1) { b = 1; break; }
if (a < b)
{
r = a; a = b; b = r;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
a = (a - b) >> 1;
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (b == 1) Jacobi = j; else Jacobi = 0;
```

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

The following line is included to the source code: if (a == 1) return 1; which also optimizes recursive Algorithm 2 and leads to recursive Algorithm 4:

Algorithm 4.

```
static long Euclid(long a, long b, ref int j)
{
long r;
if (a == 0) return b;
else
{
if ((a & 1) == 0)
{
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid(a >> 1, b, ref j);
}
if (a == 1) return 1;
if (a < b)
{
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
return Euclid(b, a, ref j);
}
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid((a - b) >> 1, b, ref j);
}
}
```

We introduce the following optimized Algorithm 5 and Algorithm 6:

Algorithm 5.

```
j = 1;
if (a < 0)
{
if ((b & 3) == 3) j = -j;
a = -a;
}
int q = 0;
if (a != 0)
{
while ((a & 1) == 0)
{
a >>= 1; r = b & 7;
```

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

```
if ((r == 3) ||(r == 5)) j = -j;
}
while (a != b)
if (a > b)
{
if (q == 1)
{
q = 0;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
a = (a - b) >> 1;
while ((a & 1) == 0)
{
a >>= 1; r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (a == 1) { b = 1; break; }
}
else
{
if (q == 0)
{
q = 1;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
r = a & 7;
if ((r == 3) ||(r == 5)) j = -j;
b = (b - a) >> 1;
while ((b & 1) == 0)
{
b >>= 1; r = a & 7;
if ((r == 3) ||(r == 5)) j = -j;
}
if (b == 1) break;
}
}
if (b == 1) Jacobi = j; else Jacobi = 0;
```

Algorithm 6.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

```
static long Euclid(long a, long b, ref int j)
{
int q = 0; long r;
if (a == 0) return b;
else
{
if ((a & 1) == 0)
{
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid(a >> 1, b, ref j);
}
if (a == 1) return 1;
}
if (a == b) return b;
else
if (a > b)
{
if (q == 1)
{
q = 0;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
r = b & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid((a - b) >> 1, b, ref j);
}
else
{
if (q == 0)
{
q = 1;
if ((a & 3) == 3 && (b & 3) == 3) j = -j;
}
r = a & 7;
if ((r == 3) ||(r == 5)) j = -j;
return Euclid((b - a) >> 1, a, ref j);
}
}
```

The recursive Algorithms 2, 4 and 6 can be called by:

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

```
j = 1;
if (a < 0) { if ((b & 3) == 3) j = -j;
a = -a; }
b = Euclid(a, b, ref j);
if (b == 1) Jacobi = j; else Jacobi = 0;
```

Numerical Example

We will test the proposed Algorithms 3, 5 and 4, 6 as well Algorithms 1 and 2 for the following example:

```
long a, b, r, d = 0;
int j, Jacobi;
for (int i = 1; i < 100000001; i++) { a = i; b = 200000002 - i;
if ((b & 1) == 0) b−−;
//Here are placed Algorithms 1, 3 and 5 and
//calling of recursive Algorithms 2, 4 and 6.
d += Jacobi; }
Console.WriteLine(d);
```

CPU time of Algorithm 1 is: **54.722 seconds.**

CPU time of Algorithm 2 is: **151.464 seconds.**

CPU time of Algorithm 3 is: **52.597 seconds.**

CPU time of Algorithm 4 is: **149.353 seconds.**

CPU time of Algorithm 5 is: **55.543 seconds.**

CPU time of Algorithm 6 is: **136.409 seconds.**

Conclusion

We present both iterative and recursive binary algorithms for finding Jacobi symbol. The presented by us Algorithms 3 and 6 possess better computational characteristics in comparison to Algorithms 1, 5 and 2, 4.

Acknowledgments

This paper is supported by the Project FP21-FMI-002 "Intelligent Innovative ICT in Research in Mathematics, Informatics and Pedagogy of Education" of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

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] T. Terzieva, *Introduction to web programming*, University Press "Paisii Hilendarski", Plovdiv (2021), ISBN 978-619-202-623-3. (in Bulgarian)

[8] T. Terzieva, *Development of algorithmic thinking in the Informatics Education*, University Press "Paisii Hilendarski", Plovdiv (2021), ISBN 978-619-202-622-6. (in Bulgarian)

[9] T. Terzieva, *Educational tools for teaching in digital environment*, University Press "Paisii Hilendarski", Plovdiv (2021). (in Bulgarian)

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

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

[18] 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)

[19] 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.

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

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

[22] 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.

[23] 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.

[24] 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.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

[25] 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.

[26] 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.

[27] 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.

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

[29] 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)

[30] 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.

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

[32] 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.

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

[34] 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.

[35] A. Iliev, N. Kyurkchiev, A. Rahnev, A New Improvement of Modular Multiplicative Inverse Binary Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 14 No. 1 (2020), 37–44.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)

[36] A. Iliev, N. Kyurkchiev, A. Rahnev, Recursive Extended Stein's Binary Algorithm, *International Electronic Journal of Pure and Applied Mathematics*, 14 No. 1 (2020), 31–36.

[37] A. Iliev, N. Kyurkchiev, A. Rahnev, A new improvement of Jacobi symbol algorithm, (2020), preprint.

[38] 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.

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

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

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

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

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

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

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

[46] J. Shallit, J. Sorenson, A Binary Algorithm for the Jacobi Symbol, *ACM SIGSAM Bulletin*, 27 No. 1 (1993), 4–11.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 15, No. 1 (2021)