International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-4, Issue-3, March 2018
ISSN: 2395-3470
www.ijseas.com

A Note on Least Absolute Remainder Euclidean Algorithm for Greatest Common Divisor

Anton Iliev[1,2], Nikolay Kyurkchiev[1,2]
[1] 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
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Bl. 8, 1113 Sofia, Bulgaria

Abstract
In this note we gave new implementation of Least Absolute Remainder Euclidean Algorithm for Greatest Common Divisor (LAREAGCD). The reason of this interest is that this is another different point of view to this problem [54, 36]. In every book of algebra and algorithms the Euclidean algorithm is part of basic examples [1-31], [33-54].Visual C# 2017 programming environment is used.
Keywords: *greatest common divisor, least absolute remainder, Euclidean Algorithm, Knuth's algorithm, reduced number of iterations.*

1. Introduction

Our work is continuation of research in [28-32].
LAREAGCD is known (see [41]):

Algorithm 1.

```
do { q = a / b; r = a - b * q;
     ar = Math.Abs(r - b); a = b;
     if (r <= ar) b = r; else b = ar; }
while (b > 0);
gcd = a;
```

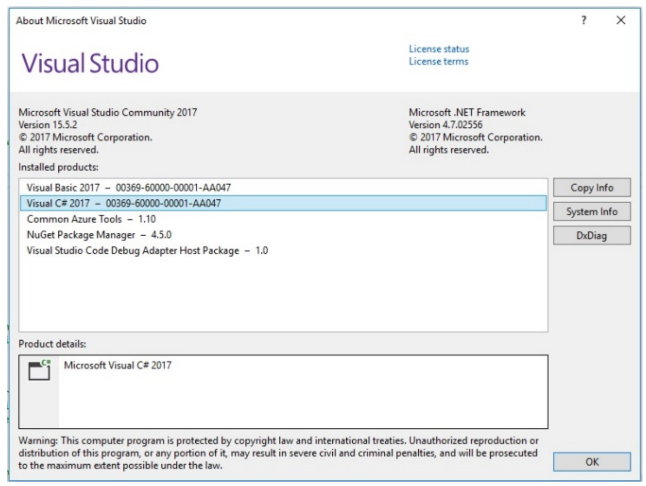

Fig. 1. Visual C# 2017

with the following programing environment (see Fig. 1.).

We present the following algorithms.

2. Main Results

Now we set the task to optimize LAREAGCD. 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.

Algorithm 2 is recursive implementation of Algorithm 1:

Algorithm 2.

```
static long Euclid(long a, long b)
    { long q = a / b; long r = a - b * q;
     long ar = Math.Abs(r - b); a = b;
     if (r <= ar) b = r; else b = ar; if (b < 1) return a;
     return Euclid(a, b); }
```

The calling of Algorithm 2 is:

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

New realization of Algorithm 1:

Algorithm 3.

```
if (a > b) do { q = a / b; r = a - b * q;
                ar = Math.Abs(r - b); a = b;
                if (r <= ar) b = r; else b = ar;
                if (b < 1) { gcd = a; break; } }
           while (true);
     else do { q = b / a; r = b - a * q;
                ar = Math.Abs(r - a); b = a;
                if (r <= ar) a = r; else a = ar;
                if (a < 1) { gcd = b; break; } }
           while (true);
```

and its recursive presentation:

Algorithm 4.

```
static long Euclid(long a, long b)
      { long q = a / b; long r = a - b * q;
        long ar = Math.Abs(r - b); a = b;
        if (r <= ar) b = r; else b = ar;
        if (b < 1) return a;
        q = b / a; r = b - a * q;
        ar = Math.Abs(r - a); b = a;
        if (r <= ar) a = r; else a = ar;
        if (a < 1) return b;
        return Euclid(b, a); }
```

And its calling:

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

3. Numerical experiments

Part 1. We will use the following task:

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

Results of Part 1.:

Time of Algorithm 1: 30.668 sec.;
Time of Algorithm 2: 48.365 sec.;
Time of Algorithm 3: 31.053 sec.;
Time of Algorithm 4: 70.053 sec.

Part 2. We will use the following task where we swapped the values of ‘a’ and ‘b’:

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

Results of Part 2.:

Time of Algorithm 1: 32.843 sec.;
Time of Algorithm 2: 49.225 sec.;
Time of Algorithm 3: 30.872 sec.;
Time of Algorithm 4: 70.111 sec.

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.

E1 = 31.7555 sec.
E2 = 48.795 sec.
E3 = 30.9625 sec.
E4 = 70.082 sec.

So you can see that our new Algorithms 3 and recursive implementation Algorithm 2 are faster than Algorithm 1 and recursive realization Algorithm 4 respectively.

Acknowledgments

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

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).
[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] G. Hardy, E. Wright, An Introduction to Theory of Numbers, 4th ed., Oxford University Press, Glasgow (1975).
[27] 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).
[28] 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.
[29] 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.

[30] 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), (accepted).

[31] 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), (accepted).

[32] 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

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

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

[35] D. E. Knuth, Evaluation of Porter’s constant, Comp. Maths. Appls., 2 (1976), 137-139.

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

[37] T. Koshy, Fibonacci and Lucas numbers with applications, John Wiley & Sons, New York (2001).

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

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

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

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

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

[43] 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.

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

[45] J. W. Porter, On a theorem of Heilbronn, Mathematika, 22 (1975), 20-28.

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

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

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

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

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

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

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

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

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