Peertechz®

Trends in Computer Science and Information Technology
mail.computerscience@peertechz.com

ENGINEERING GROUP

Trends in Computer Science and Information Technology

ISSN: 2641-3086 DOI: https://dx.doi.org/10.17352/tcsit

Received: 22 June, 2020
Accepted: 04 July, 2020
Published: 06 July, 2020

*Corresponding author: Ishmukhametov ST, Professor of Kazan Federal University, 420008, Kremlevskya st. 35, Kazan, Russia, Tel: +7 927 0304121;
E-mail: Shamil.Ishmukhametov@kpfu.ru

https://www.peertechz.com

Check for updates

Short Communication

On a new algorithm for computing GCD of integer numbers

ST Ishmukhametov[1]*, BG Mubarakov[2], RG Rubtsova[3] and Al Khalidi Arkan Mohammed[4]

[1]Professor of Kazan Federal University, 420008, Kremlevskya st. 35, Kazan, Russia

[2]Assistant professor of Kazan Federal University, 420008, Kremlevskya st. 35, Kazan, Russia

[3]postgraduate student of the Department of Computer Mathematics, Kazan Federal University, 420008, Kremlevskya st. 35 Kazan, Russia

[4]Assistant professor of Kazan Federal University, 420008, Kremlevskya st. 35 Kazan, Russia

Abstract

In the paper we give an introduction to a new algorithm counting the greatest common divisor (GCD) of natural integers called the approximating GCD algorithm introduced by S.Ishmukhametov in 2016. We compare it with the classical Euclidean GCD algorithm and the kary GCD algorithm in spirit of J. Sorenson and K. Weber and outline their advantages and disadvantages.

Introduction

The classical Euclidean GCD Algorithm is known from ancient times. Given at input two integer numbers A_o and B_o,A_o>B_o, it works at iterations called rounds implementing a single operation C_n= A_n *mod* B_n and submitting new pair $(A_{n+1}, B_{n+1}) = (B_n, C_n)$. It is very easy implemented and programmed at any programming language. Its main disadvantage is a slow convergence. If we define a convergence coefficient as $p=B/C$ then at most iterations of the Euclidean procedure it takes value less than 3. We can a little speed up the algorithm using a Lehmer's idea to unite two or more iterations into one [1].

The extended version of the Euclidean algorithm is used to solve the Besout's Equation, namely, given integers A,B find u and v, satisfying equation

$$Au+Bv=d, \qquad (1)$$

where d is the GCD of integers A and B.

Equation (1) allows us to find inverse elements by a given module, that is, to carry out the operation that has numerous applications in Number Theory and Cryptography. Namely, the module inversing is used to perform calculations in Galois Fields, generate keys for Cryptographic algorithms like RSA, DSS, carry out operations on Elliptic Curves over finite fields etc. An aacceleration of the GCD procedure can affect on a number of Cryptography algorithms.

We continue with a short description of other GCD algorithms. The binary GCD algorithm was first published by an Israeli physicist and programmer Josef Stein in 1967 [2] even it was known much earlier. His algorithm uses simpler arithmetic operations than the conventional Euclidean algorithm; it replaces division with arithmetic shifts, comparisons, and subtraction. It again consists of iterations at which given odd naturals A and B we count their difference C=|A-B| and reduce it dividing by 2 until C be odd. Then a new pair (B,C) is formed for the next iteration. The whole number of iterations is larger than in the Euclidean case but the binary algorithm outperforms the Euclidean in the operations at the algorithm step. In general, the Euclidean algorithm is preferable to the binary.

Citation: Ishmukhametov ST, Mubarakov BG, Rubtsova RG, Arkan Mohammed AK (2020) On a new algorithm for computing GCD of integer numbers. Trends Comput Sci Inf Technol 5(1): 015-017. DOI: https://dx.doi.org/10.17352/tcsit.000013

The next advantage in the area is due to Jonathan Sorenson [3], who invented the k-ary GCD algorithm that generalizes the Binary GCD algorithm. Independently, the k-ary algorithm was studied by K. Weber [4] and T. Jebelean [5].

Parameter k in this algorithm is chosen to be a comparatively small integer (usually, a power of 2 $k=2^s$). Given at input integers A and B, co-prime to k, $A>B$, the algorithm searches for integers u and v such that

$$Au+Bv \equiv o \mod k \qquad (2)$$

counts integer $C=|(Au+Bv)/k|$, and divides C by GCD(C,k) to make it co-prime to k. By

Sorenson's Theorem [3] integers u and v can be found satisfying $|u|,|v| \leq \sqrt{k}$ so coefficient $p=A/C$ xceeds $\sqrt{k}/2$. The main advantage of the k-ary algorithm is a lesser number of iterations against the Euclidean one. The larger k decreases the number of iterations but implies a longer calculation for searching s u and v. When k exceeds some optimal value, the algorithm's speed begins to fall. The direct implementation of the k-ary algorithm loses the Euclidean one but if we calculate in advance intermediate parameters like inverses by module k and store them in RAM memory of a computer we can overcome the Euclidean algorithm up to 3-5 times at integers of length 2000 or more bits (the most popular size of integers used in Cryptography).

A disadvantage of the k-ary algorithm is that the final GCD obtained by the algorithm can differ from the origin GCD. Namely, it can contain extra factors accumulated during the computation. To remove them, an additional calculation with the Euclidean GCD algorithm is performed.

In 2016 S. Ishmukhametov introduced a new approximating GCD algorithm [6] that was a modification of Sorenson's k-ary GCD algorithm. The main difference is in the choice of parameters u and v in equivalence (2). The new algorithms searches for a pair (u,v) satisfying two restrictions:

$0<u<M$ for some border M,

$|Au+Bv|$ takes a least possible value.

A	B	Euclid	K-ary (k=16)	Approx (M=16)

1485	793	C=A mod B=692	x=4, y=-4,
			C=(Ax+By)/k=173
				x=5, y=-9, C=(Ax+By)/k=18
				After reduction by 2 C=9.

Reduction *P=B/C*		1.15	4.58	88

The search of required u and v is implemented using the Farey Series [6]. Taking M=k we can reach the reduction coefficient $P=B/C$ exceeding k.

Let α be the fraction part of a real

$$\left|\frac{A/B-q}{k}\right|$$

where $q=AB^{-1} \mod k$. Farey's Series are used to find an approximation fraction m/n with limited denominator $n \leq M$ for α with minimal discrepancy $|\alpha - m/n|$. Then required (x,y) are defined as

$(x,y)=(n,f(m,n,\alpha))$,

where $f(m,n,\alpha)$ is an integer expression of arguments written in brackets.

Experimental calculations show that at input of size 500 bits and larger this algorithm is up to 5 times faster than the Euclidean Algorithm and fast versions of the k-ary GCD Algorithm. Below we give an example of GCD calculation by the Approximating Algorithm at k=64 (M=k).

	A	B	A/B	x	y	C	*P = A/C*
1	1736704041	1210259647	1,4	3	-5	13143533	132
2	1210259647	13143533	92	58	-5342	276465	4378
3	13143533	276465	48	19	-903	619	21233
4	276465	619	447	22	-9826	1	276465

The calculation finishes in 4 steps, much faster than in case of Euclidean and Sorenson's k-ary algorithms.

The extended version of the approximating GCD Algorithm was introduced in [7] by S.Ishmukahmetov, B.Mubarakov and Kamal Maad but a large amount of work was done by AlKhalidi Arkan Mohammed Ali who developed the theoretical idea of the algorithm, implemented it in the programming language C and performed experimental calculations [8,9]. Like as in the extended Euclidean Algorithm the algorithm consists of two stages. At the first stage it computes during the direct run the GCD d of input integers A and B. This stage can be performed by the Sorenson's GCsD algorithm, or using the Approximating algorithm. Then, during the inverse computation, some equations are formed to be solved using the Chinese Remainder Theorem.

The common procedure is performed up to 5 times faster than in the extended Euclidean Algorithm at integers 500 bits and larger. So this algorithm is a perspective replacement to other GCD algorithms using in the current Cryptography applications.

Conclusion

The purpose of this paper was to advertise the new approximating k-ary GCD algorithm. The algorithm works in a lesser number of iterations but with an additional Farey procedure implementing at stages of the algorithm. The direct implementation of the algorithm loses the Euclidean procedure due to a large amount of work at a stage of the algorithm. But using preliminary computed parameters of the computation we were able to overcome Euclidean's and Sorenson's algorithms up to 5 times at integers of size more than 500 bits.

Acknowledgement

This investigation was supported by grant of Scientific and Educational Mathematical Center of the Volga Federal District, agreement No. 075-02-2020-1478.

References

1. Knuth D (1998) The Art of Computer Programming. 2. **Link:** https://bit.ly/3itivu2

2. Stein J (1967) Computational problems associated with Racah algebra. Journal of Computational Physics 1: 397-405. **Link:** https://bit.ly/2D9bcrr

3. Sorenson J (1994) Two fast GCD Algorithms. J Alg 16: 110-144. **Link:** https://bit.ly/31lt9r1

4. Weber K (1995) The accelerated integer GCD algorithm. ACM Trans Math Software 21: 1-12 . **Link:** https://bit.ly/2Z07GrY

5. Jebelean T (1993) A Generalization of the Binary GCD Algorithm. Proc of Intern Symp on Symb and Algebr Comp (ISSAC'93) 111-116. **Link:** https://bit.ly/2VK2fv4

6. Ishmukhametov ST (2016) An approximating k-ary GCD Algorithm. Lobachevskii Journal of Mathematics 37: 723-729. **Link:** https://bit.ly/2ZBEMNJ

7. Ishmukhametov ST, Mubarakov BG, Al-Anni KM (2017) Calculation of Bezout's Coefficients for the k-Ary Algorithm of Finding GCD, Russian Mathematics Allerton Press Inc 61: 26-33. **Link:** https://bit.ly/2O065vD

8. Al-Khalidi Arkan M (2020) Effective Computations of Inverse by Module Elements with Approximating K-Ary GCD Algorithm by Ishmukhametov. IOP Journal of Physics 1350: 1-7. **Link:** https://bit.ly/2ZE3YTV

9. Al-Khalidi Arkan M, Ishmukhametov ST (2020) Effective programming of the GCD procedure for natural numbers, Russian Mathematics 3-8.

Discover a bigger Impact and Visibility of your article publication with Peertechz Publications

Highlights

- ❖ Signatory publisher of ORCID
- ❖ Signatory Publisher of DORA (San Francisco Declaration on Research Assessment)
- ❖ Articles archived in worlds' renowned service providers such as Portico, CNKI, AGRIS, TDNet, Base (Bielefeld University Library), CrossRef, Scilit, J-Gate etc.
- ❖ Journals indexed in ICMJE, SHERPA/ROMEO, Google Scholar etc.
- ❖ OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting)
- ❖ Dedicated Editorial Board for every journal
- ❖ Accurate and rapid peer-review process
- ❖ Increased citations of published articles through promotions
- ❖ Reduced timeline for article publication

Submit your articles and experience a new surge in publication services (https://www.peertechz.com/submission).

Peertechz journals wishes everlasting success in your every endeavours.

Copyright: © 2020 Ishmukhametov ST, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Citation: Ishmukhametov ST, Mubarakov BG, Rubtsova RG, Arkan Mohammed AK (2020) On a new algorithm for computing GCD of integer numbers. Trends Comput Sci Inf Technol 5(1): 015-017. DOI: https://dx.doi.org/10.17352/tcsit.000013