Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Analysis of the *k*-ary Euclid for tuples of integers

To cite this article: Ismail Amer and Sh T Ishmukhametov 2019 *J. Phys.: Conf. Ser.* **1352** 012001

IOP Conf. Series: Journal of Physics: Conf. Series **1352** (2019) 012001 doi:10.1088/1742-6596/1352/1/012001

Analysis of the *k*-ary Euclid for tuples of integers

Ismail Amer and Sh T Ishmukhametov

Kazan Federal University, Kazan, Russia

E-mail: safadi121979@yahoo.com

Abstract. In our paper we discuss the *k*-ary Euclid Algorithm for counting the great common divisor (GCD) of two or more integers and suggest some improvements. This gives us a possibility to parallelize and speed up the calculating of GCD, which has a variety of applications in the Number Theory, Modular Arithmetic and the Cryptography Algorithms such as RSA, ElGamal encryption system and others.

1. Introduction

Classical Euclid Algorithm is used to count the great common divisor d of two natural numbers (A, B) where $A \geq B > 0$.

It is based on a recurrent formula

$$GCD(A, B) = GCD(B, A \bmod B),$$

which is applied to diminish pairs (A, B) while the second argument is greater than 0. Then the procedure stops returning the first argument as the GCD of the origin pair (A, B).

This procedure has numerous applications in the Number Theory. Its extended version is used to count inverse elements in the finite fields. The last operation is required for generating keys of the RSA encrypting algorithm, for doubling and adding points on elliptic curves and for many others cryptographical algorithms [1].

So, even a modest acceleration of the Euclid GCD allows the researchers to save millions hours of computer time.

2. k-ary Euclid Algorithm

The *k*-ary GCD was invented by J. Sorenson (see [2, 3]). Let k be an even power of two, for example $k = 16$. Sorenson himself suggested to choose k equal to a power of a (large) prime but later Weber [4] showed that the choice $k = 2^{2s}$, where s is a natural number, is more e effective.

Let base $k = m^2$ of the algorithm chosen and $A \geq B > 0$ be integers that have no common divisors with m. The main idea of a step of the algorithm is to find some small integers x and y such that

$$Ax + By \equiv 0 \bmod k, \tag{1}$$

Then, set $C = (Ax + By)/k$ and replace the pair (A, B) by a minor pair (B, C) or (C, B) depending on if $B > C$ holds or does not. Additionally, it may require a cyclic reduction of C by two until C become odd. The algorithm is based on the following theorem:

Theorem 1. (J. Sorenson). For any $A \geq B > 0$ incomparable with $k = m^2$, there exist non-zero $x, y, |x|, |y| \leq m$ such that relation

$$Ax + By \equiv 0 \; mod \; k$$

holds.

Proof. Fix *A* and *B* such as in the theorem and let set $M = (x, y)$ to consist of all pairs (x, y) satisfying

$$-m/2 < x, y < m/2, \; x \neq 0, y \neq 0.$$

Note that the power of *M* is exactly *k*.

Define function *h* realize a map from *M* to $[0; k-1]$ as follows:

$$h(x, y) = (Ax + By) \; mod \; k.$$

Let us consider two possible cases:

1. Function *h* is injective. Then *h* performs a 1-1 map from *M* to Z_k and there exist non-zero *x, y* with $h(x, y) = 0$. Clearly, (x, y) satisfies the theorem. In this case

$$|x|, |y| \leq k/2.$$

2. Function *h* is not injective. Then there exist different pairs (x_1, y_1) and (x_2, y_2) such that $h(x_1, y_1) = h(x_2, y_2)$.

Define $x = x_1 - x_2$ and $y = y_1 - y_2$, then

$$Ax + By \; mod \; k = (Ax_1 + By_1) \; mod \; k - (Ax_2 + By_2) \; mod \; k = 0.$$

This proves the theorem.

We introduce a reduction coefficient $\rho = |A/C|$. By theorem 1, there *x* and *y* such that $\rho \geq \sqrt{k}/2 = m/2$. Indeed,

$$\rho = \frac{A}{C} = \frac{Ak}{Ax+By} \geq \frac{Ak}{2A\sqrt{k}} = \frac{\sqrt{k}}{2} \tag{2}$$

Remark. The *k*-ary GCD has a minor disadvantage that the GCD of *B* and $C = (Ax + By) \; mod \; k$ is not obligatory to be equal to $GCD(A, B)$. But it can be checked that $GCD(A, B)$ is a factor of $GCD(B, C)$. So the final value of GCD in *k*-ary method has the origin GCD as a factor. To find the origin GCD *d* we need to add at the end a final calculation:

$$d = GCD(A, d'), \; d' = \mathrm{GCD}(\mathrm{B}, \mathrm{d}''),$$

where d'' is the GCD obtained by the *k*-ary algorithm.

A search of suitable pair (x, y) *in the k-ary algorithm.*

Let $k = m^2$ for some natural $m \geq 1, m$ be even, and $A > B > 0$ be odd integers. We explain now how to choose the required *x* and *y*. The equation $Ax + By \equiv 0 \; mod \; k$ has several decisions, and the main problem is to choose a pair (x, y) with a minimal $|Ax + By|$. Since $A > B$ the decision (x, y) is better, if *x* is a small positive and *y* is negative and $y \approx rx$, where $r = -A/B$. Then additives *Ax* and *By* bilateral reduce each other.

Let $r_0 = A/B \; mod \; k, 0 < r_0 < k$. From $Ax + By \equiv 0 \; mod \; k$ we have

$y \equiv -Ax/B = -r_0 x \bmod k.$

We define $y = -r_0 x + ks,\ s \in \mathbb{Z}$ and a rational $\alpha = A/B,\ \alpha > 1$. Then

$|Ax + By| = B|\alpha x + y| = B|\alpha x - r_0 x + ks| = B|(\alpha - r_0)x + ks|,$

so our task is to find integers *y* and *s* such that function

$$d(x, s) = |(\alpha - r_0)x + ks| \tag{3}$$

takes a minimal value.

Example. $A = 12169,\ B = 583,\ k = 16$

$r_0 = \frac{A}{B} \bmod k = \frac{12169}{583} \bmod 16 = \frac{9}{7} \bmod 16 = 15,$

$\alpha = \frac{A}{B} = 20{,}9.$

$d(x, s) = |(\alpha - r_0)x + ks| = |(20{,}9 - 15)x + ks| = |5{,}9x + ks|.$

We consider two possible variants:
1. $x = 3,\ s = -1.$

$d(x, s) = |5{,}9 \cdot 3 - 16| = 1{,}7,$

$y = -r_0 x + ks = -15 \cdot 3 - 16 = -61.$

$C_1 = \left|\frac{Ax+By}{k}\right| = \left|\frac{3 \cdot 12169 - 61 \cdot 583}{16}\right| = 59.$

2. $x = 1,\ s = 0.$

$d(x, s) = |4{,}9 \cdot 1 - 0| = 4{,}9.$

$y = -r_0 x + ks = -15.$

$C_2 = \left|\frac{Ax+By}{k}\right| = \left|\frac{12169 - 15 \cdot 583}{16}\right| = 214.$

After reduction by 2 we obtain $C_2 = 107$. We see that first variant is better since $C_1 < C_2$ even the last was reduced.

3. Conclusion

In this paper, we introduced a special function $d(x, s)$ which helps us to choose at a stage of the *k*-ary algorithm best parameters *x* and *y* to increase its performance. Such modification gives an improvement not only to the *k*-ary itself, but also to the so called Weber-Jebelean algorithm. Moreover, function $d(x, s)$ helps effectively define which algorithm should be applied to current *A* and *B*, *k*-ary or Weber dmod.

Acknowledgments

This work was supported by the research Russian RFBR grant 18-47-160005.

Reference

[1] Ishmukhametov S, Mubarakov B and Mochalov A 2015 Euclidian algorithm for recurrent sequences *Applied Discrete Mathematics and Heuristic Algorithms, International Scientific Journal* (Samara: SamGU) **Vol. 1 2** 57–62
[2] Sorenson J 1990 *The k-ary GCD Algorithm* (University of Wisconsin-Madison, Tecn.Report) 1–20
[3] Sorenson J 1994 Two fast GCD Algorithms *Journal of Algorithms* **16 11** 10–144
[4] Weber K 1995 The accelerated integer GCD algorithm *ACM Trans. Math. Soft*. **21** 111–122 doi:10.1145/200979.201042