

Euclidean algorithm in Lightning-Bolt form

Jonathan L.F. King
University of Florida, Gainesville FL 32611-2082, USA
squash@ufl.edu
Webpage <http://squash.1gainesville.com/>
17 July, 2018 (at 11:44)

The Euclidean algorithm, **EU**, is often presented by a series of equations. I have found the following table-form convenient, both because it organises the computation, and gives a name to each number in the table. Because the update-rule follows the shape of a lightning-bolt, I call it the LBolt algorithm.

Henceforth, all variables are *integers* unless explicitly stated otherwise. Given integers r_0 and r_1 (for the time being, assume each is positive) we will compute $G := \text{Gcd}(r_0, r_1)$ as well as a pair S, T of **Bézout multipliers** satisfying

$$1: \quad G = S \cdot r_0 + T \cdot r_1.$$

[There is a one-parameter family of Bézout-pairs; the algorithm will compute a particular pair.] I'll explain via an example. Suppose we want the Gcd of $r_0 := 114$ and $r_1 := 33$. Then initialize the table as:

n	r_n	q_n	s_n	t_n
0	114	—	1	0
1	33		0	1

In order to compute a **BP** (Bézout-Pair), we'll need

$$1': \quad r_n = s_n \cdot 114 + t_n \cdot 33$$

to hold, for *every* n . Notice that it *already holds*, trivially, for $n=0$ and $n=1$.

At stage n , divide r_n into r_{n-1} to get a quotient, q_n , and a remainder, r_{n+1} . That is,

$$2: \quad r_{n-1} = [q_n r_n] + r_{n+1}.$$

Now use this value of q_n to update three columns:

$$3: \quad \begin{aligned} r_{n+1} &= r_{n-1} - q_n r_n; \\ s_{n+1} &:= s_{n-1} - q_n s_n; \\ t_{n+1} &:= t_{n-1} - q_n t_n. \end{aligned}$$

Doing this for $n=1$ gives

n	r_n	q_n	s_n	t_n
0	114	—	1	0
1	33	3	0	1
2	15		1	-3

Continue until you get a “0” in the r -column; I'll compute the resulting “quotient” and write “ ∞ ” in the q -column, obtaining

n	r_n	q_n	s_n	t_n
0	114	—	1	0
1	33	3	0	1
2	15	2	1	-3
3	<i>3</i>	5	<i>-2</i>	<i>7</i>
4	0	∞	11	-38

The **GCD-row** [shown here red and italicized] is the row *above* the “ ∞ -row”. The numbers we sought lie in the GCD-row. In this instance, $G = r_3$, $S = s_3$ and $T = t_3$. And indeed,

$$3 = -2 \cdot 114 + 7 \cdot 33.$$

Why the extra row? You wonder “*Why bother to compute s_4 and t_4 ?*” It isn't necessary, but they provide verification-data. Consider finding (G, S, T) when $r_0 := 98$ and $r_1 := 51$. Initialize:

n	r_n	q_n	s_n	t_n
0	98	—	1	0
1	51		0	1

Now compute...

n	r_n	q_n	s_n	t_n
0	98	—	1	0
1	51	1	0	1
2	47	1	1	-1
3	4	11	-1	2
4	3	1	12	-23
5	<i>1</i>	3	<i>-13</i>	<i>25</i>
6	0	∞	51	-98

This r_5 , which is *1*, is indeed $\text{Gcd}(98, 51)$. And

$$1 = [-13] \cdot 98 + 25 \cdot 51.$$

Now examine the ∞ -row; here, the 6th row. Note that s_6 equals r_1 upto \pm . And t_6 equals r_0 upto \pm .

In general, letting $G := \text{Gcd}(r_0, r_1)$, this “extra” row satisfies^{♥1} that

$$4: \quad s_{N+1} \cdot G = r_1 \cdot [-1]^{N+1} \quad \text{and} \quad t_{N+1} \cdot G = r_0 \cdot [-1]^N.$$

^{♥1}This is stated formally, and proven, in (9c), further below.

If you made a computational error earlier in the table, a glance at this $[N+1]^{th}$ -row will usually shout “Error!”.

Convention. Depending on context, agree to use “GCD-row” to mean both its index, and its contents. E.g, for the preceding LBolt table, expression “Let $N := \text{GCD-row}$ ” makes $N = 5$. I might also say “In the GCD-row, the t -value is 25.”

Related pamphlets. Our *Teaching page*

<http://www.math.ufl.edu/~squash/teaching.html>

has link “*practice sheet for the LBolt alg*” with pre-made tables.

There, too, is link “*Algorithms in Number Theory*” which uses LBolt iteratively to compute the $G := \text{Gcd}(M_1, M_2, \dots, M_L)$ of a *list* of integers, computing also Bézout multipliers S_1, S_2, \dots, S_L st.

$$5: \quad \sum_{\ell=1}^L S_\ell M_\ell = G.$$

We call $\vec{S} := (S_1, \dots, S_L)$ a *Bézout tuple* for the given tuple $\vec{M} := (M_1, \dots, M_L)$.

Exer: Fix an L -tuple \vec{M} which is not the all-zero tuple. Prove that the set of Bézout tuples for \vec{M} is $[L-1]$ -dimensional.

The 2nd page of “*Algorithms in NT*” describes an algorithm for solving linear congruences such as $33x \equiv_{114} 18$, and has a worked-example.

Proving the Euclidean Alg. works

I’ll leave this as an **Exer: The Euclidean-Alg always halts.**

Define the divisor and common-divisor sets,

$$\mathcal{D}(K) := \{d \in \mathbb{Z} \mid d \blacktriangleright K\} \quad \text{and}$$

$$\mathcal{C}(K, N) := \mathcal{D}(K) \cap \mathcal{D}(N).$$

[Below, “LC” stands for “Linear Combination”.]

6: LC Lemma. Consider integers α, β, γ, M such that

$$6a: \quad \alpha + [M \cdot \beta] = \gamma.$$

Then

$$*: \quad \mathcal{C}(\alpha, \beta) = \mathcal{C}(\beta, \gamma). \quad \blacklozenge$$

Proof. Each $d \in \mathcal{C}(\alpha, \beta)$ necessarily divides $\alpha + [M\beta]$, since $M \in \mathbb{Z}$. Thus $\mathcal{C}(\alpha, \beta) \subset \mathcal{D}(\gamma)$. By its definition, $\mathcal{C}(\alpha, \beta) \subset \mathcal{D}(\beta)$. Consequently

$$6b: \quad \mathcal{C}(\alpha, \beta) \subset \mathcal{C}(\beta, \gamma).$$

OTOHand, we can rewrite (6a) as

$$\gamma + [-M \cdot \beta] = \alpha.$$

The above reasoning hands us

$$6c: \quad \mathcal{C}(\alpha, \beta) \supset \mathcal{C}(\beta, \gamma).$$

This, together with (6b), yields (*). ◆

6d: Corollary. Consider an LBolt seeded with integers r_0 and r_1 . Then $\mathcal{C}(r_k, r_{k+1}) = \mathcal{C}(r_0, r_1)$, for each index k . Consequently,

$$6e: \quad \text{Gcd}(r_k, r_{k+1}) = \text{Gcd}(r_0, r_1).$$

Letting N be the GCD-row index, then,

$$6f: \quad r_N = \text{Gcd}(r_0, r_1),$$

since r_{N+1} is zero. ◆

7: Bézout Lemma. Consider an LBolt seeded with integers r_0 and r_1 . For each k , then,

$$\mathbf{B}(k): \quad r_k = [s_k r_0] + [t_k r_1]$$

holds. I'll refer to assertion $[\forall k \in \mathbb{N}: \mathbf{B}(k)]$ as the **Bézout row-property** or **LBolt row-property**.

With $N := \text{GCD-index}$, consequently,

$$7a: \quad \text{Gcd}(r_0, r_1) = [s_N r_0] + [t_N r_1]. \quad \diamond$$

Proof. The LBolt-seeding gives $\mathbf{B}(0)$ and $\mathbf{B}(1)$. Now fix a posint n st. $\mathbf{B}(n-1)$ and $\mathbf{B}(n)$. Courtesy (3),

$$\begin{aligned} & s_{n+1}r_0 + t_{n+1}r_1 \\ &= [[s_{n-1} - \mathfrak{q}_n s_n] \cdot r_0] + [[t_{n-1} - \mathfrak{q}_n t_n] \cdot r_1] \\ &= [s_{n-1}r_0 + t_{n-1}r_1] - \mathfrak{q}_n \cdot [s_n r_0 + t_n r_1], \end{aligned}$$

since multiplication distributes-over addition. Assertions $\mathbf{B}(n-1)$ and $\mathbf{B}(n)$ now give us that

$$s_{n+1}r_0 + t_{n+1}r_1 = [r_{n-1}] - \mathfrak{q}_n \cdot [r_n]$$

which, by (3) again, equals r_{n+1} . We've thus inductively established

$$\forall k \geq 1: \quad [\mathbf{B}(k-1) \ \& \ \mathbf{B}(k)] \implies \mathbf{B}(k+1). \quad \blacklozenge$$

Alternate initialization

Consider an LBolt seeded with integers r_0 and r_1 . Define matrices

$$M_n := \begin{bmatrix} s_n & t_n \\ s_{n+1} & t_{n+1} \end{bmatrix} \quad \text{and} \quad R_n := \begin{bmatrix} r_n \\ r_{n+1} \end{bmatrix}.$$

Up till now, our initialization matrix M_0 has the identity matrix $\mathbf{I} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. However, our Bézout Lemma proof only used $\mathbf{B}(0)$ and $\mathbf{B}(1)$, i.e that

*: $M_0 \cdot R_0 = R_0.$

and so other values of M_0 are possible.

As an example, the usual LBolt for $\text{Gcd}(3, 2)$ is

8a:

n	r_n	q_n	s_n	t_n
0	3	—	1	0
1	2	1	0	1
2	1	2	1	-1
3	0	∞	-2	3

Another initial-matrix is $M_0 := \begin{bmatrix} 3 & -3 \\ 0 & 1 \end{bmatrix}$, yielding

8b:

n	r_n	q_n	s_n	t_n
0	3	—	3	-3
1	2	1	0	1
2	1	2	3	-4
3	0	∞	-6	9

Row-2 gives us a *different* Bézout pair. We might conjecture that check-pair $(-6, 9)$ equals the check-pair $(-2, 3)$ from the first table, but multiplied by $\text{Det}(M_0)$.

Yet another init-matrix is $M_0 := \begin{bmatrix} 7 & -9 \\ 2 & -2 \end{bmatrix}$, producing

8c:

n	r_n	q_n	s_n	t_n
0	3	—	7	-9
1	2	1	2	-2
2	1	2	5	-7
3	0	∞	-8	12

Row-2 gives us a *third* Bézout pair. And the check-pair $(-8, 12)$ indeed equals $\text{Det}(\begin{bmatrix} 7 & -9 \\ 2 & -2 \end{bmatrix})$ times the $(-2, 3)$ from our first table.

Check-row. We study an LBolt seeded with a co-prime pair $r_0 \perp r_1$, and initial-matrix M_0 st. (*) holds.

For $k \geq 1$, let

$$Q_k := \begin{bmatrix} 0 & 1 \\ 1 & -q_k \end{bmatrix}$$

and observe $\text{Det}(Q_k) = -1$. Define product matrix

$$P_n := Q_n \cdots Q_2 Q_1;$$

hence P_0 is the identity matrix \mathbf{I} .

Update-rule (3) tells us that

$$R_k = Q_k \cdot R_{k-1} \quad \text{and} \quad M_k = Q_k \cdot M_{k-1}.$$

Consequently,

9a: $R_n = P_n \cdot R_0, \quad M_n = P_n \cdot M_0$
and $\text{Det}(M_n) = [-1]^n \cdot \text{Det}(M_0).$

Moreover,

9b: $M_n \cdot R_0 = P_n M_0 \cdot R_0 \xrightarrow{\text{by (*)}} P_n R_0 = R_n.$

Letting $N := \text{GCD-index}$, we have that

** : $M_N \cdot R_0 = R_N \xrightarrow{\text{recall}} \begin{bmatrix} 1 \\ 0 \end{bmatrix},$

since $r_0 \perp r_1$. Have $(S, T) := (s_N, t_N)$ denote the Bézout-pair, and let $(\alpha, \beta) := (s_{N+1}, t_{N+1})$ be the pair whose values we wish to determine. Finally, set $\delta := \text{Det}(M_0)$.

Our (**) gives the top two lines of

$$\begin{aligned} Sr_0 + Tr_1 &= 1, \\ \alpha r_0 + \beta r_1 &= 0. \quad \text{Notice that} \\ -\alpha T + \beta S &= \delta \cdot [-1]^N \end{aligned}$$

follows by computing $\text{Det}(M_N)$. Multiplying the middle eqn by T and the bottom by r_0 gives

$$\begin{aligned} \alpha T r_0 + \beta T r_1 &= 0 \quad \text{and} \\ -\alpha T r_0 + \beta S r_0 &= r_0 \delta [-1]^N. \end{aligned}$$

Adding them yields

$$\beta \stackrel{\text{note}}{=} \beta \cdot [Sr_0 + Tr_1] = r_0 \delta \cdot [-1]^N.$$

Finally, plugging this into the middle eqn gives

$$0 = \alpha r_0 + r_0 \delta [-1]^N \cdot r_1.$$

When $r_0 \neq 0$, then $0 = \alpha + r_1 \delta [-1]^N$. Hence

$$\alpha = -r_1\delta \cdot [-1]^N = r_1\delta \cdot [-1]^{N+1}.$$

We have proven the following theorem.

9c: Check-value Theorem. Consider an LBolt seeded with integers $r_0 \neq 0$ and r_1 , together with an initial-matrix M_0 satisfying

$$9d: \quad M_0 \cdot \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \end{bmatrix}.$$

Let $N := \text{GCD-index}$ and $G := \text{Gcd}(r_0, r_1)$. Then

$$9e: \quad \begin{aligned} s_{N+1} \cdot G &= r_1 \cdot \text{Det}(M_0) \cdot [-1]^{N+1} \\ \text{and } t_{N+1} \cdot G &= r_0 \cdot \text{Det}(M_0) \cdot [-1]^N. \end{aligned}$$

(Recall that our standard LBolt has $\text{Det}(M_0) = 1$.) \diamond

Filename: Problems/NumberTheory/lightning-bolt.tex
As of: Thursday 06Jan2011. Typeset: 17Jul2018 at 11:44.