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/
12 February, 2021 (at *23:30*)

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 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 $\mathcal{G} := \text{GCD}(r_0, r_1)$ as well as a pair S,T of ***Bézout multipliers*** satisfying

$$1: \qquad \mathcal{G} \quad = \quad S \cdot r_0 + T \cdot r_1 \,.$$

[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

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

to hold, for <u>*every*</u> 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,

$$2: \qquad r_{n-1} \quad = \quad [\mathsf{q}_n r_n] + r_{n+1} \,.$$

Now use this value of q_n to update three columns:

$$3: \qquad \begin{aligned} r_{n+1} &= r_{n-1} - \mathsf{q}_n r_n \,; \\ s_{n+1} &:= s_{n-1} - \mathsf{q}_n s_n \,; \\ t_{n+1} &:= t_{n-1} - \mathsf{q}_n t_n \,. \end{aligned}$$

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	*3*	5	*-2*	*7*
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, $\mathcal{G} = r_3$, $S = s_3$ and $T = t_3$. And indeed,

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

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 $(\mathcal{G}, S, T)$ 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	*1*	3	*-13*	*25*
6	0	∞	51	-98

This r_5, which is *1*, is indeed GCD(98, 51). And

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

Now examine the ∞-row; here, the 6$^{\text{th}}$ row. Note that s_6 equals r_1 upto ±. And t_6 equals r_0 upto ±.

In general, letting $\mathcal{G} := \text{GCD}(r_0, r_1)$, this "extra" row satisfies[♡1] that

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

[♡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 := $ 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 $\mathcal{G} := \text{GCD}(M_1, M_2, \dots, M_L)$ of a *list* of integers, computing also a Bézout multipliers $S_1, S_2, \dots, S_L$ st.

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

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

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

The 2nd page of "*Algorithms in NT*" describes an algorithm for solving linear congruences such as $33{\cdot}x \equiv_{114} 18$, 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,

$$\mathcal{D}(K) := \left\{ d \in \mathbb{Z} \;\middle|\; d \bullet\!| \; K \right\} \quad \text{and}$$
$$\mathcal{C}(K, N) := \mathcal{D}(K) \cap \mathcal{D}(N)\,.$$

[Below, "LC" stands for "Linear Combination".]

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

6a: $$\alpha + [M{\cdot}\beta] = \gamma\,.$$

Then

∗: $$\mathcal{C}(\alpha, \beta) = \mathcal{C}(\beta, \gamma)\,.$$ ♢

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

OTOHand, we can rewrite (6a) as

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

The above reasoning hands us

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

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: $$\text{GCD}(r_k, r_{k+1}) = \text{GCD}(r_0, r_1)\,.$$

Letting N be the GCD-row index, then,

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

since r_{N+1} is zero. ♢

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

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

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

With $N :=$ GCD-index, consequently,

$$\text{7a:} \qquad \text{GCD}(r_0, r_1) = [s_N r_0] + [t_N r_1]\,. \qquad \diamondsuit$$

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 update rule (3),

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

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

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

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

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

Alternate initialization

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

$$\mathsf{M}_n := \begin{bmatrix} s_n & t_n \\ s_{n+1} & t_{n+1} \end{bmatrix} \quad\text{and}\quad \mathsf{R}_n := \begin{bmatrix} r_n \\ r_{n+1} \end{bmatrix}.$$

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

$$*: \qquad \mathsf{M}_0 \cdot \mathsf{R}_0 = \mathsf{R}_0 .$$

and so other values of M_0 are possible.

As an example, the usual LBolt for $\mathrm{GCD}(3,2)$ 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 $\mathsf{M}_0 := \begin{bmatrix} 3 & -3 \\ 0 & 1 \end{bmatrix}$, 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 $\mathrm{Det}(\mathsf{M}_0)$.

Yet another init-matrix is $\mathsf{M}_0 := \begin{bmatrix} 7 & -9 \\ 2 & -2 \end{bmatrix}$, 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. The check-pair $(-8,12)$ indeed equals $\mathrm{Det}\left(\begin{smallmatrix} 7 & -9 \\ 2 & -2 \end{smallmatrix}\right)$ times the $(-2,3)$ from our first table.

In this last example

8d:

n	r_n	q_n	s_n	t_n
0	3	▬	-5	9
1	2	1	-4	7
2	1	2	-1	2
3	0	∞	-2	3

has $\mathsf{M}_0 := \begin{bmatrix} -5 & 9 \\ -4 & 7 \end{bmatrix}$, whose determinant is 1, hence yielding the same check-tuple $(-2,3)$ as in table (8a).

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

For $k \geqslant 1$, let

$$\mathsf{Q}_k := \begin{bmatrix} 0 & 1 \\ 1 & -\mathsf{q}_k \end{bmatrix}$$

and observe $\mathrm{Det}(\mathsf{Q}_k) = \text{-}1$. Define product matrix

$$\mathsf{P}_n := \mathsf{Q}_n \cdots \mathsf{Q}_2 \mathsf{Q}_1 ;$$

hence P_0, the empty product, is the identity matrix $\mathbf{I}$.

Update-rule (3) tells us that

$$\mathsf{R}_k = \mathsf{Q}_k \cdot \mathsf{R}_{k-1} \quad \text{and} \quad \mathsf{M}_k = \mathsf{Q}_k \cdot \mathsf{M}_{k-1} .$$

Consequently,

9a:
$$\mathsf{R}_n = \mathsf{P}_n \cdot \mathsf{R}_0 , \qquad \mathsf{M}_n = \mathsf{P}_n \cdot \mathsf{M}_0$$
$$\text{and} \quad \mathrm{Det}(\mathsf{M}_n) = \mathrm{Det}(\mathsf{M}_0) \cdot [\text{-}1]^n .$$

Moreover,

9b:
$$\mathsf{M}_n \cdot \mathsf{R}_0 = \mathsf{P}_n \mathsf{M}_0 \cdot \mathsf{R}_0 \stackrel{\text{by } (*)}{=\!=} \mathsf{P}_n \mathsf{R}_0 = \mathsf{R}_n .$$

Letting $N :=$ GCD-index, we have that

$**$:
$$\mathsf{M}_N \cdot \mathsf{R}_0 = \mathsf{R}_N \stackrel{\text{recall}}{=\!=} \begin{bmatrix} 1 \\ 0 \end{bmatrix} ,$$

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 $\boldsymbol{\delta} := \mathrm{Det}(\mathsf{M}_0)$.

Our $(**)$ gives the top two lines of

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

courtesy (9a), since $\mathrm{Det}(\mathsf{M}_N) = \mathrm{Det}\left(\begin{bmatrix} S & T \\ \alpha & \beta \end{bmatrix}\right)$. Multiplying the middle eqn by T and the bottom by r_0 gives

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

Adding them yields

$$\beta \stackrel{\text{note}}{=\!=} \beta \cdot [S r_0 + T r_1] = r_0 \, \boldsymbol{\delta} \cdot [\text{-}1]^N .$$

Finally, plugging this into the middle eqn gives

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

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

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

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:
$$\mathsf{M}_0 \cdot \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} .$$

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

9e:
$$\begin{aligned} s_{N+1} \cdot \mathcal{G} &= r_1 \cdot \mathrm{Det}(\mathsf{M}_0) \cdot [\text{-}1]^{N+1} \\ \text{and} \quad t_{N+1} \cdot \mathcal{G} &= r_0 \cdot \mathrm{Det}(\mathsf{M}_0) \cdot [\text{-}1]^N . \end{aligned}$$

(*Recall that our standard LBolt has* $\mathrm{Det}(\mathsf{M}_0) = 1$.) ♢

Filename: `Problems/NumberTheory/lightning-bolt.latex`
As of: *Thursday 06Jan2011.* Typeset: *12Feb2021 at 23:30.*