Greatest Common Divisors, the Euclidean Algorithm and Bézout's Lemma

Ryan C. Daileda

Trinity University

Number Theory

Introduction

One of the most important results in the theory of divisibility in $\mathbb{Z}$ is the Fundamental Theorem of Arithmetic (FTA).

The FTA asserts that every natural number (greater than 1) can be expressed uniquely as a product of prime numbers.

Crucial to proving the FTA is a result known as Euclid's Lemma (which we won't state here).

Euclid's Lemma, in turn, is an easy consequence of what we will call Bézout's Lemma, which is concerned with greatest common divisors.

And that is where we will begin...

Common Divisors

Given $a, b \in \mathbb{Z}$, let

$$C(a, b) = \{d \in \mathbb{N} : d|a \text{ and } d|b\},$$

the set of *(positive) common divisors* of a and b.

The set $C(a, b)$ is never empty since $1 \in C(a, b)$.

Furthermore, if a and b are both nonzero,

$$d \in C(a, b) \quad \Rightarrow \quad d \leq \min\{|a|, |b|\},$$

by the properties of divisibility discussed earlier.

On the other hand, since $d|0$ for all $d \in \mathbb{N}$, $C(0, b)$ is just the set of (positive) divisors of b.

So, if $b \neq 0$, then

$$d \in C(0, b) \ \Rightarrow \ d \leq |b|.$$

Since $C(a, b) = C(b, a)$, we have proven that as long as a and b are not *both* zero, $C(a, b)$ is a nonempty *finite* set of positive integers.

Note that $C(0, 0) = \mathbb{N}$, however.

The Greatest Common Divisor

Definition

Let $a, b \in \mathbb{Z}$, not both zero. The *greatest common divisor* (GCD) of a and b, denoted (a, b), is the largest element of $C(a, b)$:

$$(a, b) = \max C(a, b).$$

Because $C(a, b) \subset \mathbb{N}$ is finite and nonempty when a and b are not both 0, (a, b) is a well-defined positive integer.

Although it may seem counterintuitive, it will be convenient to define $(0, 0) = 0$.

Examples

- We have
$$(8,76)=4,\ \ (91,70)=7,\ \ (72,84)=12,$$
$$(54,39)=3,\ \ (16,69)=1.$$

- For all $a,b\in\mathbb{Z}$,
$$(a,b)=(b,a).$$

- For any $a\in\mathbb{Z}$,
$$(a,0)=|a|.$$

Computing the GCD

Let $a, b \in \mathbb{Z}$ both be nonzero.

We now pose our main question: how can (a, b) be computed?

One option is brute force: perform trial divisions by every positive $d \leq \min\{|a|, |b|\}$ to compute $C(a, b)$ explicitly.

Although this process must end in a finite number of steps, it is extremely inefficient.

We can derive a much more efficient procedure based on the following observation.

Periodicity of the GCD

Lemma 1

Let $a, b \in \mathbb{Z}$. For any $n \in \mathbb{Z}$

$$(a, b) = (a, b + na).$$

Remark. Lemma 1 tells us that, as a function of b, the GCD (a, b) is periodic with period a.

Proof of Lemma 1. If $a = 0$, there is nothing to prove, so we may assume $a \neq 0$.

It therefore suffices to prove that $C(a, b) = C(a, b + na)$.

Let $d \in C(a, b)$.

Because $d|a$ and $d|b$, d divides $b + na$ since it is a linear combination of a and b.

Thus $d \in C(a, b + na)$. This proves that $C(a, b) \subseteq C(a, b + na)$.

Now suppose $d \in C(a, b + na)$. Then $d|a$ and $d|b + na$, so that d also divides the linear combination

$$(-n)a + (b + na) = b.$$

Therefore $d \in C(a, b)$. This shows that $C(a, b + na) \subseteq C(a, b)$ and completes the proof. □

GCDs and the Division Algorithm

We can now connect GCDs to the Division Algorithm.

Corollary 1

Let $a, b \in \mathbb{Z}$ with $a \neq 0$. Use the Division Algorithm to write $b = qa + r$ with $0 \leq r < |a|$. Then

$$(a, b) = (a, r).$$

Proof. According to Lemma 1 we have

$$(a, b) = (a, qa + r) = (a, r).$$

□

So, when faced with a (nontrivial) GCD (a, b), by performing a division we can always assume b is smaller than a.

But $(a, b) = (b, a)$...

Example. Consider the GCD $(91, 70)$. By Corollary 1 we have

$$(91, 70) = (70, 91) = (70, 21) = (21, 70) = (21, 7) = 7.$$

This procedure is the basis of the *Euclidean Algorithm* for computing the GCD.

The Euclidean Algorithm (EA)

Let $a, b \in \mathbb{Z}$ be nonzero. Consider the following sequence of divisions and GCD equalities:

$$\begin{aligned} b &= q_1 a + r_1 & \quad (a, b) &= (a, r_1) \\ a &= q_2 r_1 + r_2 & (r_1, a) &= (r_1, r_2) \\ r_1 &= q_3 r_2 + r_3 & (r_2, r_1) &= (r_2, r_3) \\ &\vdots & &\vdots \\ r_{k-1} &= q_{k+1} r_k + r_{k+1} & (r_k, r_{k-1}) &= (r_k, r_{k+1}) \\ &\vdots & &\vdots \end{aligned}$$

in which the remainders satisfy

$$|a| > r_1 > r_2 > r_3 > \cdots > r_k > r_{k+1} > \cdots \geq 0.$$

Because the remainders r_k are nonnegative integers, they cannot decrease indefinitely.

Therefore the sequence eventually terminates after $n+1$ divisions with

$$r_{n-1} = q_{n+1} r_n,$$

i.e. $r_{n+1} = 0$, so that the final GCD equation reads

$$(r_n, r_{n-1}) = (r_n, 0) = r_n.$$

Because the GCD remains unchanged at every stage, this means that

$$(a, b) = r_n.$$

That is, the final nonzero remainder will be (a, b)!

Example

Let's compute $(336, 726)$. We have

$$\begin{aligned} 726 &= 2 \cdot 336 + 54, \\ 336 &= 6 \cdot 54 + 12, \\ 54 &= 4 \cdot 12 + 6, \\ 12 &= 2 \cdot 6. \end{aligned}$$

Since the last nonzero remainder is 6, it must be the case that

$$\boxed{(336, 726) = 6.}$$

Remark. Note that we have found the GCD without any prior knowledge of the divisors of either 336 or 726.

A Different Point of View

Strictly speaking, in terms of computing the GCD, the quotients in the Euclidean Algorithm serve no purpose.

It is the equality of GCDs of pairs of remainders that make the algorithm valid.

However, if we analyze the algorithm from a different perspective, we will find that the quotients contain "hidden" information about (a, b) and its relationship to a and b.

Use the remainders in the Euclidean Algorithm to form the vectors

$$\mathbf{x}_0 = \begin{pmatrix} b \\ a \end{pmatrix}, \quad \mathbf{x}_1 = \begin{pmatrix} a \\ r_1 \end{pmatrix}, \quad \mathbf{x}_k = \begin{pmatrix} r_{k-1} \\ r_k \end{pmatrix} \quad \text{for } k \geq 2.$$

Let

$$Q_k = \begin{pmatrix} 0 & 1 \\ 1 & -q_k \end{pmatrix} \quad \text{for } k \geq 1.$$

Notice that we can re-express the equalities of the EA as

$$\begin{aligned} \mathbf{x}_{k+1} = \begin{pmatrix} r_k \\ r_{k+1} \end{pmatrix} &= \begin{pmatrix} r_k \\ r_{k-1} - q_{k+1} r_k \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & -q_{k+1} \end{pmatrix} \begin{pmatrix} r_{k-1} \\ r_k \end{pmatrix} = Q_{k+1} \mathbf{x}_k, \end{aligned}$$

for $k \geq 0$.

Therefore, if we work backward, we obtain

$$\mathbf{x}_n = Q_n \mathbf{x}_{n-1} = Q_n Q_{n-1} \mathbf{x}_{n-2} = \cdots = Q_n Q_{n-1} \cdots Q_1 \mathbf{x}_0.$$

Because $r_n = (a, b)$, this is equivalent to

$$Q_n Q_{n-1} \cdots Q_1 \begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} * \\ (a, b) \end{pmatrix}. \tag{1}$$

Finally, write

$$Q_n Q_{n-1} \cdots Q_1 = \begin{pmatrix} * & * \\ s & r \end{pmatrix}, \tag{2}$$

where $r, s \in \mathbb{Z}$ since the Q_k are integral matrices.

Substituting (2) into (1) and comparing the bottom entry on both sides, we find that a, b, r, s satisfy the relationship

$$ra + sb = (a, b).$$

We have just given a constructive proof of *Bézout's Lemma*.

Theorem 1 (Bézout's Lemma)

Let $a, b \in \mathbb{Z}$. There exist $r, s \in \mathbb{Z}$ so that

$$(a, b) = ra + sb.$$

As we have just seen, the quotients in the Euclidean Algorithm can be used to compute r and s explicitly.

Remarks

- Although the existence of r and s in Bézout's Lemma is primarily of theoretical importance, r and s do have practical applications, so it's handy to have a way to find them.
- The values of r and s are not unique, but we will give a complete description when we study linear Diophantine equations.
- Note that when computing r and s from the quotients q_k, the final quotient q_{n+1} is *not used*.

Example

Applied to $(a, b) = (336, 726) = 6$, the Euclidean Algorithm took $n + 1 = 4$ divisions, so we need the first $n = 3$ quotients, which are

$$q_1 = 2, \quad q_2 = 6, \quad q_3 = 4.$$

We have

$$Q_3 Q_2 Q_1 = \begin{pmatrix} 0 & 1 \\ 1 & -4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -6 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} = \begin{pmatrix} -6 & 13 \\ 25 & -54 \end{pmatrix}.$$

Thus we can take $r = -54$, $s = 25$ in Bézout's Lemma. That is, we have

$$-54 \cdot 336 + 25 \cdot 726 = 6.$$

Efficiency of the Euclidean Algorithm

So just how efficient is the EA? Specifically, how many steps (divisions) do we expect it to take?

The answer is related to the Fibonacci sequence $\{F_n\}$:

$$F_1 = F_2 = 1, \;\; F_{n+2} = F_{n+1} + F_n \text{ for } n \geq 1.$$

Specifically, one can prove:

Theorem 2

Let $a, b \in \mathbb{N}$ with $a < b$. Let N be the largest index so that $F_N \leq b$. Then the EA takes at most $N - 2$ steps to compute (a, b), and this bound is sharp.

Recall that the Fibonacci numbers are given explicitly by

$$F_n = \frac{1}{\sqrt{5}}\left(\phi^n - (\phi^*)^n\right),$$

where

$$\phi = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad \phi^* = \frac{1-\sqrt{5}}{2}$$

are the golden ratio and its algebraic conjugate.

Since $|\phi^*| < 1$, this means $F_n \approx \frac{\phi^n}{\sqrt{5}}$ for large n.

So, for large b, we will have $F_N \leq b$ (roughly) if and only if $\frac{\phi^N}{\sqrt{5}} \leq b$, or $N \leq \log(b\sqrt{5})/\log(\phi)$.

Therefore the number of steps in the EA is asymptotically logarithmic (at worst) in the larger input.