For example, the common divisors of 30 and 24 are $-6, -3, -2, -1, 1, 2, 3,$ and 6.

ition 35.2 **(Greatest common divisor)** Let $a, b \in \mathbb{Z}$. We call an integer d the *greatest common divisor* of a and b provided

1. d is a common divisor of a and b and
2. if e is a common divisor of a and b, then $e \leq d$.

The greatest common divisor of a and b is denoted $\gcd(a, b)$.

For example, the greatest common divisor of 30 and 24 is 6, and we write $\gcd(30, 24) = 6$. Also $\gcd(-30, -24) = 6$.

Nearly every pair of integers has a greatest common divisor (see Exercise 35.3), and if a and b have a gcd, it is unique (Exercise 35.5). This justifies our use of the definite article when we call $\gcd(a, b)$ *the* greatest common divisor of a and b.

In this section, we explore the various properties of greatest common divisors.

Calculating the gcd

precisely ional

In the foregoing example, we calculated the greatest common divisor of 30 and 24 by explicitly listing all their common factors and choosing the largest. This suggests an *algorithm* for computing gcd. The algorithm is as follows:

- Suppose a and b are positive integers.
- For every positive integer k from 1 to the smaller of a and b, see whether $k|a$ and $k|b$. If so, save that number k on a list.
- Choose the largest number on the list. That number is $\gcd(a, b)$.

This procedure works: Given any two positive integers a and b, it finds their gcd. However, it is a dreadful algorithm because even for moderately large numbers (e.g., $a = 34902$ and $b = 34299883$), the algorithm needs to do many, many divisions. So although correct, this algorithm is terribly slow.

There is a clever way to calculate the greatest common divisor of two positive integers; this procedure was invented by Euclid. It is not only very fast, but it is not difficult to implement as a computer program.

The central idea in Euclid's Algorithm is the following result.

35.3 Let a and b be positive integers and let $c = a \bmod b$. Then

$$\gcd(a, b) = \gcd(b, c).$$

In other words, for positive integers a and b, we have

$$\gcd(a, b) = \gcd(b, a \bmod b).$$

Proof. We are given that $c = a \bmod b$. This means that $a = qb + c$ where $0 \leq c < b$.

Let $d = \gcd(a, b)$ and let $e = \gcd(b, c)$. Our goal is to prove that $d = e$. To do this, we prove that $d \leq e$ and $d \geq e$.

First, we show $d \leq e$. Since $d = \gcd(a, b)$, we know that $d|a$ and $d|b$. We can write $c = a - qb$. Since a and b are multiples of d, so is c. Thus d is a common divisor of b and c. However, e is the greatest common divisor of b and c, so $d \leq e$.

Next, we show $d \geq e$. Since $e = \gcd(b, c)$, we know that $e|b$ and $e|c$. Now $a = qb + c$, and hence $e|a$ as well. Since $e|a$ and $e|b$, we see that e is a common divisor of a and b. However, d is the greatest common divisor of a and b, so $d \geq e$.

We have shown $d \leq e$ and $d \geq e$, and hence $d = e$; that is, $\gcd(a, b) = \gcd(b, c)$.

■

To illustrate how Proposition 35.3 enables us to calculate greatest common divisors efficiently, we compute $\gcd(689, 234)$. The simple, inefficient divide-and-check algorithm we considered first would have us try all possible common divisors from 1 to 234 and select the largest. This implies we would perform $234 \times 2 = 468$ division problems!

Instead, we use Proposition 35.3. To find $\gcd(689, 234)$, let $a = 689$ and $b = 234$. We find $c = 689 \bmod 234$. This requires us to do a division. The result is $c = 221$. To find $\gcd(689, 234)$, it is enough to find $\gcd(234, 221)$ because these two values are the same. Let's record this step here:

$$689 \bmod 234 = 221 \qquad \Rightarrow \qquad \gcd(689, 234) = \gcd(234, 221).$$

Now all we have to do is calculate $\gcd(234, 221)$. We use the same idea. We apply Proposition 35.3 as follows. To find $\gcd(234, 221)$, we calculate $234 \bmod 221 = 13$. Thus $\gcd(234, 221) = \gcd(221, 13)$. Let's record this step (division #2).

$$234 \bmod 221 = 13 \qquad \Rightarrow \qquad \gcd(234, 221) = \gcd(221, 13).$$

Now the problem is reduced to $\gcd(221, 13)$. Notice that the numbers are significantly smaller than the original 689 and 234. We again use Proposition 35.3 and calculate $221 \bmod 13 = 0$. What does that mean? It means that when we divide 221 by 13, there is no remainder. In other words, $13|221$. So clearly the greatest common divisor of 221 and 13 is 13. Let's record this step (division #3).

$$221 \bmod 13 = 0 \qquad \Rightarrow \qquad \gcd(221, 13) = 13.$$

We are finished! We have done three divisions (not 468 ☺), and we found

$$\gcd(689, 234) = \gcd(234, 221) = \gcd(221, 13) = 13.$$

The steps we just performed are precisely the Euclidean algorithm. Here is a formal description:

Euclid's Algorithm for Greatest Common Divisor

Input: Positive integers a and b.
Output: $\gcd(a, b)$.

- Let $c = a \bmod b$.
- If $c = 0$, then we return the answer b and stop.
- Otherwise ($c \neq 0$), we calculate $\gcd(b, c)$ and return this as the answer.

This algorithm for gcd is defined in terms of itself. This is an example of a *recursively* defined algorithm (see Exercise 21.8, where recursion is explored). Let's see how the algorithm works for the integers $a = 63$ and $b = 75$.

- The first step is to calculate $c = a \bmod b$, and we get $c = 63 \bmod 75 = 63$.
- Next we check whether $c = 0$. It's not, so we go on to compute $\gcd(b, c) = \gcd(75, 63)$.

  Very little progress has been made so far! All the algorithm has done is reverse the numbers. The next pass through, however, is more interesting.
- Now we restart the process with $a' = 75$ and $b' = 63$. We calculate $c' = 75 \bmod 63 = 12$. Since $12 \neq 0$, we are told to calculate $\gcd(b', c') = \gcd(63, 12)$.
- We restart again with $a'' = 63$ and $b'' = 12$. We calculate $c'' = 63 \bmod 12 = 3$. Since this is not zero, we need to go on and to calculate $\gcd(b'', c'') = \gcd(12, 3)$.
- We restart yet again with $a''' = 12$ and $b''' = 3$. Now we are told to calculate $c''' = 12 \bmod 3 = 0$. Aha! Now $c''' = 0$, so we return the answer $b''' = 3$ and we are finished.

  The final answer is that $\gcd(63, 75) = 3$.

Here is an overview of the calculation in chart form:

a	b	c
63	75	63
75	63	12
63	12	3
12	3	0

With only four divisions, the answer is produced.

Here is another way to visualize this computation. We create a list whose first two entries are a and b. Now we extend the list by computing mod of the last two entries of the list. When we reach 0, we stop. The next-to-last entry is the gcd of a and b. In this example, the list would be

$$(63, 75, 63, 12, 3, 0).$$

Correctness

Just because someone writes down a procedure to calculate gcd does not make it correct. The point of mathematics is to prove its assertions; the correctness of an algorithm is no exception.

n 35.4 **(Correctness of Euclid's Algorithm for gcd)** Euclid's Algorithm correctly computes $\gcd(a, b)$ for any positive integers a and b.

Proof. Suppose, for the sake of contradiction, that Euclid's Algorithm did not correctly compute gcd. Then there is some pair of positive integers a and b for which it fails. Choose a and b such that $a+b$ is as small as possible. (We are using the smallest-counterexample method.)

It might be the case that $a < b$. If this is so, then the first pass through Euclid's Algorithm will simply interchange the values a and b [as we saw when we calculated $\gcd(63, 75)$] because if $a < b$ then $c = a \bmod b = a$, and Euclid's Algorithm directs us to calculate $\gcd(b, c) = \gcd(b, a)$.

Thus we may assume that $a \geq b$.

The first step of the algorithm is to calculate $c = \gcd(a, b)$. Two outcomes are possible: either $c = 0$ or $c \neq 0$.

In the case $c = 0$, $a \bmod b = 0$, which implies $b|a$. Since b is the largest divisor of b (since $b > 0$ by hypothesis) and since $b|a$, we have b is the greatest common divisor of a and b. In other words, the algorithm gives the correct result, contradicting our supposition that it fails for a and b.

So it must be the case that $c \neq 0$. To get c, we calculated the remainder when dividing a by b. By Theorem 34.1, we have $a = qb + c$ where $0 < c < b$. We also know that $b \leq a$. We add the inequalities:

$$\begin{array}{rc} & c < b \\ + & b \leq a \\ \Rightarrow & b + c < a + b \end{array}$$

Thus b, c are positive integers with $b + c < a + b$.

This means that b and c are not a counterexample to the correctness of Euclid's Algorithm because $b + c < a + b$, and among all counterexamples, a and b was a counterexample with the smallest sum. Thus the algorithm correctly computes $\gcd(b, c)$ and returns its value as the answer. However, by Proposition 35.3, this is the right answer! This contradicts the supposition that Euclid's Algorithm fails on a, b.$\Rightarrow\Leftarrow$ Hence Euclid's Algorithm always returns the greatest common divisor of the positive integers it is given. ■

How Fast?

How many times do we have to divide to calculate the greatest common divisor of two positive integers? We claim that after two rounds of Euclid's Algorithm, the integers with which we are working have decreased by at least 50%. This is the main tool.

5.5 Let $a, b \in \mathbb{Z}$ with $a \geq b > 0$. Let $c = a \bmod b$. Then $c < \frac{a}{2}$.

Proof. We consider two cases: (1) $a < 2b$ and (2) $a \geq 2b$.

- **Case (1):** $a < 2b$.

  We know that $2b > a > 0$, so $a > 0$ and $a - b \geq 0$, but $a -$
  Hence the quotient when a is divided by b is 1. So the remainder in a
  by b is $c = a - b$.

  Now we can rewrite $a < 2b$ as $b > \frac{a}{2}$, and so

  $$c = a - b < a - \frac{a}{2} = \frac{a}{2}$$

  which is what we wanted.
- **Case (2):** $a \geq 2b$, which can be rewritten $b \leq \frac{a}{2}$.

  The remainder, upon division of a by b, is less than b. So $c < b$,
  have $b \leq \frac{a}{2}$, so $c < \frac{a}{2}$.

In both cases, we found $c < \frac{a}{2}$.

We may assume that we start Euclid's Algorithm with $a \geq b$; if not, th
rithm reverses a and b on its first pass, and from there on, the numbers com
creasing order. That is, if the numbers produced by Euclid's Algorithm are l

$$(a, b, c, d, e, f, \ldots, 0)$$

then, assuming $a \geq b$, we have

$$a \geq b \geq c \geq d \geq e \geq f \geq \cdots \geq 0.$$

By Proposition 35.5, the numbers c and d are less than half as large as a
respectively. Likewise, two steps later, the numbers e and f are less than
large as c and d, respectively, and less than one-fourth of a and b, respectivel

> *Every two steps of Euclid's Algorithm decreases the integers with which v*
> *are working to less than half their current values.*

If we begin with (a, b), then two steps later, the numbers are less than $(\frac{1}{2}a$
and four steps later, less than $(\frac{1}{4}a, \frac{1}{4}b)$, and six steps later, less than $(\frac{1}{8}a, \frac{1}{8}b)$
large are the numbers after $2t$ passes of Euclid's Algorithm? Since every twc
decrease the numbers by more than a factor of 2, we know that after $2t$ ste
numbers drop by more than a factor of 2^t; that is, the two numbers are les
$(2^{-t}a, 2^{-t}b)$.

Euclid's Algorithm stops when the second number reaches zero. Sinc
numbers in Euclid's Algorithm are integers, this is the same as when the se
number is less than 1. This means that as soon as we have

$$2^{-t}b \leq 1,$$

the second number must have reached zero. Taking base-2 logs of both side
have

$$\begin{aligned} \log_2[2^{-t}b] &\leq \log_2 1 \\ -t + \log_2 b &\leq 0 \\ \log_2 b &\leq t. \end{aligned}$$

In other words, once $t \geq \log_2 b$, the algorithm must be finished. So after $2\log_2$ passes, the algorithm has completed its work.

How many divisions might this be if, say, a and b were enormous numbe (e.g., 1000 digits each). If $b \approx 10^{1000}$, then the number of steps is bounded by

$$2\log_2(10^{1000}) = 2000\log_2 10 < 2000 \times 3.4 = 6800.$$

(*Note*: $\log_2 10 \approx 3.3219 < 3.4$.) So in under 7000 steps, we have our answe Compare this to doing 10^{1000} divisions (see Exercise 35.8)!

I hope you do not think I am trying your patience by considering such ridiculous example. Why on earth would anyone want to compute the gcd of tw 1000-digit numbers!? Well, the fact is that this is a practical, important problen with both industrial and military applications. More on this later!

An Important Theorem

The following theorem is central to the study of the greatest common divisor (and beyond).

orem 35.6 Let a and b be integers, not both zero. The smallest positive integer of the form $ax + by$, where x and y are integers, is $\gcd(a, b)$.

e integers. An *combination* ıny number $x + by$ where so integers. tells us that

er linear f a and b is

For example, suppose $a = 30$ and $b = 24$. We can make a chart of the values $ax + by$ for integers x and y between -4 and 4. We get the following table:

		y								
		-4	-3	-2	-1	0	1	2	3	4
	-4	-216	-192	-168	-144	-120	-96	-72	-48	-24
	-3	-186	-162	-138	-114	-90	-66	-42	-18	6
	-2	-156	-132	-108	-84	-60	-36	-12	12	36
	-1	-126	-102	-78	-54	-30	-6	18	42	66
x	0	-96	-72	-48	-24	0	24	48	72	96
	1	-66	-42	-18	6	30	54	78	102	126
	2	-36	-12	12	36	60	84	108	132	156
	3	-6	18	42	66	90	114	138	162	186
	4	24	48	72	96	120	144	168	192	216

What is the smallest positive value on this chart? We see the number 6 at $x = -3$, $y = 4$ (because $30 \times -3 + 24 \times 4 = -90 + 96 = 6$) and again at $x = 1$, $y = -1$ (because $30 \times 1 + 24 \times -1 = 30 - 24 = 6$).

Now we have shown only a relatively small portion of all the possible values of $ax + by$. Is it possible, if we were to extend this chart, that we might find a smaller positive value for $30x + 24y$? The answer is no. Notice that both 30 and 24 are divisible by 6. Therefore any integer of the form $30x + 24y$ is also divisible by 6 (see Exercise 4.8). So even if we extended this chart