Lecture 15: Modular Inverses

Eric Vigoda
Discrete Mathematics for Computer Science

March 4, 2026

15 Multiplicative Inverses and Euclid's Algorithm

In the last lecture we learned about modular arithmetic and the basic arithmetic operations of addition/subtraction and multiplication. We also looked at the fast exponentation algorithm using repeated squaring. If we have n-bit numbers x, y, and N, then $x^y \bmod N$ can be computed in time polynomial in n, even though the numbers are potentially as large as 2^n. This will be extremely important in our upcoming cryptography applications as n will be a few thousand bits long, so polynomial in n is reasonable, but an algorithm which is exponential in n will finish after the end of the universe.

The next question is what about division? What is the analog of division in modular arithmetic. We begin with multiplicative inverses, which are a fascinating topic and the key to much of the beautiful mathematics in modular arithmetic. We'll soon see that Euclid's algorithm (which is often referred to as the Euclidean algorithm) is a key algorithmic tool in this context.

15.1 Multiplicative Inverses: Definition

What is the integer x where

$$3x \equiv 1 \bmod 16$$

By trying the integers between 0 and 15 we see that for $x = 11$ then

$$3x \equiv 33 \equiv 1 \bmod 16$$

In standard arithmetic, when we solve $3y = 1$ then $y = 3^{-1} = 1/3$ and is referred to as the inverse of 3. Of course $1/3$ doesn't make any sense when we're working **mod 16** as the only valid numbers are integers (and in simplest terms it's integers between 0 and 15). Nevertheless, we refer to 11 as the multiplicative inverse of 3 modulo 16.

Notice that if we ask for the y where

$$11y \equiv 1 \bmod 16$$

then $y = 3$ satisfies this equation, and hence 3 is the multiplicative inverse of 11 modulo 16. So it appears multiplicative inverses are symmetric. In particular, inverses always come in pairs (except when a number is its own inverse).

Let's look at a few more illustrative examples before formalizing inverses. What is the integer z where

$$2z \equiv 1 \bmod 16$$

By trying all integers between 0 and 15 we observe that there is no such z. Why is that? We'll explore in a moment when a number a has an inverse modulo N, and if it does have an inverse then how do we find it.

We begin by formalizing the notion of an inverse in modular arithmetic.

Definition 15.1. For integer a and integer $N \geq 1$, we say that x is the multiplicative inverse of a modulo N if the following holds:

$$ax \equiv 1 \bmod N$$

And we denote such an x by

$$x \equiv a^{-1} \bmod N$$

An important property is that the symmetry of multiplicative inverses, if they exist. By the definition of multiplicative inverses, x is the multiplicative inverse of a modulo N iff a is the multiplicative inverse of x modulo N:

$$x \equiv a^{-1} \bmod N \iff a \equiv x^{-1} \bmod N;$$

this is because both statements are the same as $ax \equiv 1 \bmod N$.

15.2 Inverses: Existence

Let's look at an example of multiplicative inverses. Let $N = 13$ and let's consider the inverses modulo 13:

$1^{-1} \equiv 1 \bmod 13$	note, 1 is always it's own inverse since $1^2 = 1$
$2^{-1} \equiv 7 \bmod 13$	since $2 \times 7 = 14 = 13 + 1$
$3^{-1} \equiv 9 \bmod 13$	since $3 \times 9 = 27 = 13 \times 2 + 1$
$4^{-1} \equiv 10 \bmod 13$	since $4 \times 10 = 40 = 13 \times 3 + 1$
$5^{-1} \equiv 8 \bmod 13$	since $5 \times 8 = 40 = 13 \times 3 + 1$
$6^{-1} \equiv 11 \bmod 13$	since $6 \times 11 = 66 = 13 \times 5 + 1$
$7^{-1} \equiv 2 \bmod 13$	since $2^{-1} \equiv 7 \bmod 13$
$8^{-1} \equiv 5 \bmod 13$	since $5^{-1} \equiv 8 \bmod 13$
$9^{-1} \equiv 3 \bmod 13$	since $3^{-1} \equiv 9 \bmod 13$
$10^{-1} \equiv 4 \bmod 13$	since $4^{-1} \equiv 10 \bmod 13$
$11^{-1} \equiv 6 \bmod 13$	since $6^{-1} \equiv 11 \bmod 13$
$12^{-1} \equiv 12 \bmod 13$	note, -1 is always it's own inverse since $(-1)^2 = 1$

Now let's consider the example of $N = 12$, where we'll use the notation DNE for does not exist when there is no multiplicative inverse (which we can determine by trying all possibilities between 1 and 11):

$$1^{-1} \equiv 1 \bmod 12$$
$$2^{-1} \bmod 12 \text{ DNE}$$
$$3^{-1} \bmod 12 \text{ DNE}$$
$$4^{-1} \bmod 12 \text{ DNE}$$
$$5^{-1} \equiv 5 \bmod 12$$
$$6^{-1} \bmod 12 \text{ DNE}$$
$$7^{-1} \equiv 7 \bmod 12$$
$$8^{-1} \bmod 12 \text{ DNE}$$
$$9^{-1} \bmod 12 \text{ DNE}$$
$$10^{-1} \bmod 12 \text{ DNE}$$
$$11^{-1} \equiv 11 \bmod 12$$

Something interesting is happening here: why do 5 and 7 have inverses modulo 12 but $2, 3, 4, 6, 8, 9, 10$ do not have inverses? I left out 1 and 11 since these always have inverses (they are self-inverses). And why did every number have an inverse modulo 13?

There are two questions:

When does an inverse exist?
And if it exists, how do we compute it efficiently?

Well, our first observation is that 13 is a prime number. Then we notice that that 5 and 7 are **relatively prime** to 12 which means: $gcd(5, 12) = 1$ and $gcd(7, 12) = 1$; whereas the other numbers have a common factor with 12 so for $x \in \{2, 3, 4, 6, 8, 9, 10\}$ then $gcd(x, 12) > 1$.

Saying a pair of numbers x and y are relatively prime means that x behaves like a prime when we work modulo y, and similarly y behaves like a prime when we work modulo x. More formally, we say x and y are **relatively prime** iff $gcd(x, y) = 1$ (and thus they share no common factors).

The key property is that $a^{-1} \bmod N$ exists iff a and N are relatively prime, which means $gcd(a, N) = 1$.

Lemma 15.2. *For integers a and N where $a \geq 1$ and $N \geq 1$, then*

$$a^{-1} \bmod N \text{ exists} \iff gcd(a, N) = 1,$$

and similarly, $N^{-1} \bmod a$ exists if and only if $gcd(a, N) = 1$.

In the case that $gcd(a, N) = 1$ then recall that it's symmetric so $a^{-1} \equiv x \bmod N$ if and only if $x^{-1} \equiv a \bmod N$.

We'll prove one direction of the lemma: if $gcd(a, N) > 1$ then $a^{-1} \bmod N$ DNE. To prove the other direction, which states that if $gcd(a, N) = 1$ then $a^{-1} \bmod N$ exists, we'll show an algorithm to find the inverse when $gcd(a, N) = 1$.

Suppose $d = gcd(a, N) \geq 2$ then why is there no inverse of $a^{-1} \bmod N$? Let's consider the simpler case where a and N are both even, so 2 is a common divisor of a and N (there may be bigger divisors as well). Recall,

$$x \equiv a^{-1} \bmod N \iff ax \equiv 1 \bmod N \iff ax = kN + 1 \text{ for some integer } k.$$

We're trying to show there is no such x. Since a is even then ax is also even for any integer x; that means $ax = 2\ell$ for some integer ℓ. Similarly, since N is even then $N = 2j$ for some integer j. Thus, for any integer k,

$$ax - kN = 2\ell - 2kj = 2(\ell - kj).$$

Since ℓ, k, j are all integers then $\ell - kj$ is an integer, and thus $ax - kN$ is an even number since it's equal to 2 times some integer (namely, $\ell - kj$). But for x to be an inverse of a modulo N then we need that $ax - kN = 1$ but that is clearly not true. Similarly, given any common divisor $d > 1$ of a and N then we end up with that $ax - kN$ is a multiple of d and therefore $ax - kN \neq 1$ so $ax \not\equiv 1 \bmod N$, which proves this direction (the negative direction) of the lemma.

15.3 Inverses: Uniqueness

Before proceeding with the algorithm for finding the inverse let us first observe that if the inverse exists then it's unique (there is only one possible choice).

Lemma 15.3 (Uniqueness of Multiplicative Inverses)**.** *If a multiplicative inverse of a modulo N exists, then it is unique modulo N.*

This means that if $a^{-1} \bmod N$ exists then there is exactly one number in $1, \ldots, N-1$ which is the inverse and hence it's a unique number.

Lemma 15.4 (Uniqueness of Multiplicative Inverses)**.** *If a multiplicative inverse of a modulo N exists, then it is unique modulo N.*

Proof. Suppose x and y are both multiplicative inverses of a modulo N. Then

$$ax \equiv 1 \pmod{N} \qquad \text{and} \qquad ay \equiv 1 \pmod{N}.$$

Hence

$$ax \equiv ay \pmod{N}.$$

Subtracting the two expressions gives

$$a(x - y) \equiv 0 \pmod{N}.$$

Since x is an inverse of a, we have $ax \equiv 1 \pmod{N}$. Multiplying both sides of the congruence above by x yields

$$x\,a(x - y) \equiv x \cdot 0 \pmod{N}.$$

Using associativity and the fact that $ax \equiv 1 \pmod{N}$, we obtain

$$1 \cdot (x - y) \equiv 0 \pmod{N},$$

and therefore

$$x - y \equiv 0 \pmod{N}.$$

Thus

$$x \equiv y \pmod{N},$$

which shows that the multiplicative inverse is unique modulo N. □

The key step in the above proof was that since $a^{-1} \bmod N$ exists then we can multiply both sides of the equation by $a^{-1} \bmod N$ and this simplifies the equation.

The uniqueness of the inverse explains why it's clear to simply write $a^{-1} \bmod N$, since there is only one possible inverse (up to equivalence modulo N). We have now proved that if $\gcd(a, N) > 1$ then no inverse exists. Later, using the Extended Euclid algorithm, we will prove the converse: if $\gcd(a, N) = 1$, then an inverse can be efficiently computed.

15.4 Euclid's GCD Algorithm

We now know exactly when inverses exist. The remaining question is algorithmic:

Given a and N with $\gcd(a, N) = 1$, how do we efficiently compute the inverse?

To answer this, we first need an efficient method for computing greatest common divisors.

For integers $x, y \geq 0$, let $\gcd(x, y)$ denote the greatest common divisor of x and y. Note, $\gcd(x, y) = \max\{d : d|x, d|y\}$. What if $x = 0$ and/or $y = 0$? We consider every number z as a divisor of 0 (since $0 = kz$ for some integer k, namely $k = 0$). Since the gcd must divide both numbers, the common divisors of x and 0 are exactly the divisors of x. Hence, the largest one is x, and therefore $\gcd(x, 0) = x$ for integer $x > 0$. What about $\gcd(0, 0)$? That we have to define, and we'll define it as $\gcd(0, 0) = 0$.

The basis of Euclid's GCD algorithm is the following fact.

Lemma 15.5 (Euclid's Rule)**.** *For integers x, y with $x \geq y > 0$,*

$$\gcd(x, y) = \gcd(y, x \bmod y).$$

First, observe that since $x \bmod y \leq y - 1 < y$, this is why we ordered the operands in the RHS as $y \geq x \bmod y$.

Let's first illustrate the lemma. Suppose $x = 323$ and $y = 51$ then the lemma says that $\gcd(323, 51) = \gcd(51, 17)$, which greatly simplifies the calculation as in this case it is easy to see that $\gcd(51, 17) = 3$.

The above lemma (Euclid's Rule) follows from the following elementary fact:

$$\gcd(x, y) = \gcd(x - y,\ y).$$

To see why this fact holds, we need to show two directions. In the forward direction, if d is a divisor of x and y, then d is a divisor of $x - y$ and y; that proves that $\gcd(x, y) \leq \gcd(x - y, y)$. In the reverse direction, if d is a divisor of $x - y$ and y, then d is a divisor of x and y; that proves that $\gcd(x - y, y) \leq \gcd(x, y)$. Together, these two directions show that $\gcd(x, y) = \gcd(x - y, y)$ as claimed.

Let's begin with the forward direction. Suppose that d is a divisor of x and y. That means there exist integers k and ℓ where $x = kd$ and $y = \ell d$. Thus, $x - y = kd - \ell d = d(k - \ell) = dj$ where $j = k - \ell$ is an integer since k and ℓ are integers. Hence, d is a divisor of $x - y$ and also y, which completes the proof of the forward direction.

For the reverse direction, suppose d is a divisor of $x - y$ and y, and hence $x - y = kd$ and $y = \ell d$ for integers k, ℓ. Therefore, $x = (x - y) + y = kd + \ell d = d(k + \ell) = dj$ where $j = k + \ell$ is an integer, which proves the reverse direction and completes the proof of this fact that $\gcd(x, y) = \gcd(x - y,\ y)$.

We now explain why this implies Euclid's Rule. Write

$$x = qy + r \qquad \text{where} \qquad q = \left\lfloor \frac{x}{y} \right\rfloor \quad \text{and} \quad r = x \bmod y.$$

Since replacing x by $x - y$ does not change the greatest common divisor, we may repeatedly subtract y from x without affecting the gcd. After subtracting y exactly q times, we obtain

$$x - qy = r.$$

Therefore,

$$\gcd(x, y) = \gcd(r, y) = \gcd(y, r),$$

which proves Euclid's Rule:

$$\gcd(x, y) = \gcd(y, x \bmod y).$$

Euclid's Rule leads to an easy recursive algorithm.

Euclid(x, y) :

Input: integers x, y with $x \geq y \geq 0$

Output: $\gcd(x, y)$

- if $y = 0$ then return(x).
- else return(**Euclid**$(y,\ x \bmod y)$).

15.5 Running Time Analysis

Observation 15.6. If $x \geq y > 0$, then

$$x \bmod y < \frac{x}{2}.$$

Proof. There are two cases.

Case 1: $y \leq \frac{x}{2}$.
Then

$$x \bmod y < y \leq \frac{x}{2}.$$

Case 2: $y > \frac{x}{2}$.
Then dividing x by y gives quotient 1, so

$$x \bmod y = x - y < \frac{x}{2}.$$

In both cases,

$$x \bmod y < \frac{x}{2}.$$

□

Theorem 15.7. *If x and y are n-bit integers, then Euclid's algorithm performs at most $2n$ recursive calls.*

Proof. Starting from input (x, y), we then get $(y, x \bmod y)$, and then $(x \bmod y, z)$ for some z. By the above observation we know that $x \bmod y < x/2$. Hence, the first argument (which is the larger of the two inputs) decreases by at least a factor of 2 after at most two recursive calls. Halving an integer strictly reduces its binary length by at least one bit. Therefore, after two recursive calls the number of bits in the first input decreases, and hence for an initial n-bit number there are at most $2n$ recursive calls. □

Total Running Time. Each recursive call requires computing $x \bmod y$. Since division of two n-bit integers takes $O(n^2)$ time (using standard multiplication/division algorithms), and there are at most $O(n)$ recursive calls, then the total running time is

$$O(n) \cdot O(n^2) = O(n^3).$$

Thus, Euclid's algorithm runs in time polynomial in n, which is the number of bits of the input.

Example: Euclid's Algorithm in Action

Let us compute

$$\gcd(1071, 462).$$

We repeatedly apply Euclid's Rule:

$$\gcd(x, y) = \gcd(y, x \bmod y).$$

$$\begin{aligned}\gcd(1071, 462) &= \gcd(462, 1071 \bmod 462)\\ &= \gcd(462, 147)\end{aligned}$$

$$\begin{aligned}\gcd(462, 147) &= \gcd(147, 462 \bmod 147)\\ &= \gcd(147, 21)\end{aligned}$$

$$\begin{aligned}\gcd(147, 21) &= \gcd(21, 147 \bmod 21)\\ &= \gcd(21, 0).\end{aligned}$$

Since $\gcd(a, 0) = a$, we conclude

$$\gcd(1071, 462) = 21.$$

15.6 Computing Inverses: Extended Euclid Algorithm

We now have Euclid's GCD algorithm to efficiently compute the GCD of two integers x and y. If $\gcd(x, y) = 1$ then we'd also like to compute their inverses, namely, $x^{-1} \bmod y$ and $y^{-1} \bmod x$. To do that we make a small but nontrivial modification to Euclid's algorithm.

In Euclid's GCD algorithm, the input is a pair of integers x and y, and the output is an integer d where $d = \gcd(x, y)$. In the Extended Euclid algorithm, the input will again be a pair x, y, and the output will now be a triple (d, a, b) where $d = \gcd(x, y)$ and a, b are integers where $d = ax + by$. Thus this is a generalization of Euclid's GCD algorithm.

Why is it interesting to obtain a pair a, b where $d = ax + by$? If $d = 1$ so $\gcd(x, y) = 1$ then $ax + by = 1$. That means:

$$ax + by \equiv 1 \bmod y, \text{ and thus } ax \equiv 1 \bmod y.$$

In other words,

$$x^{-1} \equiv a \bmod y,$$

and so a is the inverse of x modulo y. Similarly, b is the inverse of y modulo x. Therefore, when $d = \gcd(x, y) = 1$ then the Extended Euclid algorithm returns the inverse of x modulo y, and also the inverse of y modulo x.

The outputs a and b will be integers but they will not be in simplified form, in particular one of the two will typically be a negative number. This is a byproduct of how the algorithm operates.

Extended-Euclid(x, y) :

Input: integers x, y with $x \geq y \geq 0$

Output: (d, a, b) where $d = \gcd(x, y)$ and $d = ax + by$ for integers a and b.

- if $y = 0$ then return$(x, 1, 0)$.
- $(d, a', b') =$ **Extended-Euclid**$(y, x \bmod y)$
- Let $q = \lfloor \frac{x}{y} \rfloor$.
- Return$(d, b', a' - qb')$

The proof of correctness is by strong induction and involves a bit messy algebra so we skip it in these notes, and refer the interested reader to Chapter 1 of the Algorithms textbook [DPV06]. We note that the first coordinate d of the output is the same as the original Euclid's GCD algorithm so that is certainly correct, as before.

We show an illustration of the Extended-Euclid algorithm to find the inverses in a non-trivial example.

15.7 Example of Extended Euclid Algorithm

We will compute the following inverse:

$$911^{-1} \pmod{1769}.$$

First note that $\gcd(911, 1769) = 1$ (we will verify this during the algorithm), so an inverse of 911 modulo 1769 exists.

Step 1: Trace the recursion down

We list the recursive calls, where each call is of the form

$$\text{Extended-Euclid}(x, y) \;\to\; \text{Extended-Euclid}(y,\; x \bmod y).$$

$$\begin{aligned}
\text{EE}(1769, 911) &\to \text{EE}(911, 1769 \bmod 911) = \text{EE}(911, 858)\\
\text{EE}(911, 858) &\to \text{EE}(858, 911 \bmod 858) = \text{EE}(858, 53)\\
\text{EE}(858, 53) &\to \text{EE}(53, 858 \bmod 53) = \text{EE}(53, 10)\\
\text{EE}(53, 10) &\to \text{EE}(10, 53 \bmod 10) = \text{EE}(10, 3)\\
\text{EE}(10, 3) &\to \text{EE}(3, 10 \bmod 3) = \text{EE}(3, 1)\\
\text{EE}(3, 1) &\to \text{EE}(1, 3 \bmod 1) = \text{EE}(1, 0).
\end{aligned}$$

Base case:

$$\text{EE}(1, 0) = (d, a, b) = (1, 1, 0),$$

which is correct since $1 = 1 \cdot 1 + 0 \cdot 0$.

Step 2: Trace the outputs back up

Now we back-substitute, recording (d, a, b) for each call where $q = \left\lfloor \frac{x}{y} \right\rfloor$ and the return value is

$$(d, b', a' - qb').$$

$$\begin{aligned}
\text{EE}(1, 0) &: (1, 1, 0) && \\
\text{EE}(3, 1) &: q = 3,\ (1, 0, 1 - 3 \cdot 0) && \Rightarrow (1, 0, 1)\\
\text{EE}(10, 3) &: q = 3,\ (1, 1, 0 - 3 \cdot 1) && \Rightarrow (1, 1, -3)\\
\text{EE}(53, 10) &: q = 5,\ (1, -3, 1 - 5 \cdot (-3)) && \Rightarrow (1, -3, 16)\\
\text{EE}(858, 53) &: q = 16,\ (1, 16, -3 - 16 \cdot 16) && \Rightarrow (1, 16, -259)\\
\text{EE}(911, 858) &: q = 1,\ (1, -259, 16 - 1 \cdot (-259)) && \Rightarrow (1, -259, 275)\\
\text{EE}(1769, 911) &: q = 1,\ (1, 275, -259 - 1 \cdot 275) && \Rightarrow (1, 275, -534).
\end{aligned}$$

Thus we have expressed:

$$1 = 275 \cdot 1769 + (-534) \cdot 911.$$

Notice that the coefficient of 911 is -534, which is the modular inverse of 911 modulo 1769.

Step 3: Extract the inverse

Reducing both sides modulo 1769, we eliminate the multiple of 1769:

$$1 \equiv (-534) \cdot 911 \pmod{1769}.$$

Therefore,

$$911^{-1} \equiv -534 \pmod{1769}.$$

To present it in simplified terms we have the following:

$$911^{-1} \equiv -534 \equiv 1769 - 534 \equiv 1235 \pmod{1769}.$$

In conclusion, the multiplicative inverse of 911 modulo 1769 is 1235:

$$911^{-1} \equiv 1235 \pmod{1769}.$$

To verify, we see that $911 \cdot 1235 \equiv 1 \pmod{1769}$.

References

[DPV06] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill, 2006.