Preview

We are building the tools for learning about the RSA cryptosystem—soon!

Preview

We are building the tools for learning about the RSA cryptosystem—soon!

Today: Building the foundations of modular arithmetic.

Review

- Say $x \equiv y \pmod{m}$ if $m \mid x - y$.

Review

- Say $x \equiv y \pmod{m}$ if $m \mid x - y$.
- If $a \equiv c \pmod{m}$ and $b \equiv d \pmod{m}$, then $a + b \equiv c + d \pmod{m}$ and $ab \equiv cd \pmod{m}$.

Review

- Say $x \equiv y \pmod{m}$ if $m \mid x - y$.
- If $a \equiv c \pmod{m}$ and $b \equiv d \pmod{m}$, then $a + b \equiv c + d \pmod{m}$ and $ab \equiv cd \pmod{m}$.
- Notation: $\mathbb{Z}/m\mathbb{Z} = \{0, 1, \ldots, m - 1\}$ with the operations of addition and multiplication modulo m.

Review

- Say $x \equiv y \pmod{m}$ if $m \mid x - y$.
- If $a \equiv c \pmod{m}$ and $b \equiv d \pmod{m}$, then $a + b \equiv c + d \pmod{m}$ and $ab \equiv cd \pmod{m}$.
- Notation: $\mathbb{Z}/m\mathbb{Z} = \{0, 1, \ldots, m-1\}$ with the operations of addition and multiplication modulo m.
- Each $a \in \mathbb{Z}$ has a unique representative in $\{0, 1, \ldots, m-1\}$.

Review

- Say $x \equiv y \pmod{m}$ if $m \mid x - y$.
- If $a \equiv c \pmod{m}$ and $b \equiv d \pmod{m}$, then $a + b \equiv c + d \pmod{m}$ and $ab \equiv cd \pmod{m}$.
- Notation: $\mathbb{Z}/m\mathbb{Z} = \{0, 1, \ldots, m-1\}$ with the operations of addition and multiplication modulo m.
- Each $a \in \mathbb{Z}$ has a unique representative in $\{0, 1, \ldots, m-1\}$.
- For $a \in \mathbb{Z}/m\mathbb{Z}$, a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$ if and only if $\gcd(a, m) = 1$.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$?

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$?

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on…

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on...

Thus we are led to study the map $f(x) = ax \pmod{m}$ as x ranges over $\mathbb{Z}/m\mathbb{Z}$.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on...

Thus we are led to study the map $f(x) = ax \pmod{m}$ as x ranges over $\mathbb{Z}/m\mathbb{Z}$.

Insight: If $\gcd(a, m) \neq 1$, then the map f sends some non-zero elements to zero.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on...

Thus we are led to study the map $f(x) = ax \pmod{m}$ as x ranges over $\mathbb{Z}/m\mathbb{Z}$.

Insight: If $\gcd(a, m) \neq 1$, then the map f sends some non-zero elements to zero.

- Example: Multiplication by 3, modulo 6.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on...

Thus we are led to study the map $f(x) = ax \pmod{m}$ as x ranges over $\mathbb{Z}/m\mathbb{Z}$.

Insight: If $\gcd(a, m) \neq 1$, then the map f sends some non-zero elements to zero.

- Example: Multiplication by 3, modulo 6.
- This means 3 cannot have an inverse modulo 6.

Review of Multiplicative Inverses

Say $a \in \mathbb{Z}/m\mathbb{Z}$. How might we look for a^{-1}?

Try every possibility.

- Is $a^{-1} = 1$? Check if $a \cdot 1 \equiv 1 \pmod{m}$.
- Is $a^{-1} = 2$? Check if $a \cdot 2 \equiv 1 \pmod{m}$.
- So on...

Thus we are led to study the map $f(x) = ax \pmod{m}$ as x ranges over $\mathbb{Z}/m\mathbb{Z}$.

Insight: If $\gcd(a, m) \neq 1$, then the map f sends some non-zero elements to zero.

- Example: Multiplication by 3, modulo 6.
- This means 3 cannot have an inverse modulo 6.

On the other hand, if $\gcd(a, m) = 1$, then f is bijective, which gives us our inverse.

Greatest Common Divisor

For two integers $a, b \in \mathbb{Z}$, the **greatest common divisor (GCD)** of a and b is the largest number that divides both a and b.

Greatest Common Divisor

For two integers $a, b \in \mathbb{Z}$, the **greatest common divisor (GCD)** of a and b is the largest number that divides both a and b.

Fact: Any common divisor of a and b also divides $\gcd(a, b)$.

Greatest Common Divisor

For two integers $a, b \in \mathbb{Z}$, the **greatest common divisor (GCD)** of a and b is the largest number that divides both a and b.

Fact: Any common divisor of a and b also divides $\gcd(a, b)$.

- If not, then d has a prime factor that $\gcd(a, b)$ does not.

Greatest Common Divisor

For two integers $a, b \in \mathbb{Z}$, the **greatest common divisor (GCD)** of a and b is the largest number that divides both a and b.

Fact: Any common divisor of a and b also divides $\gcd(a, b)$.

- If not, then d has a prime factor that $\gcd(a, b)$ does not.
- This prime factor p divides both a and b.

Greatest Common Divisor

For two integers $a, b \in \mathbb{Z}$, the **greatest common divisor (GCD)** of a and b is the largest number that divides both a and b.

Fact: Any common divisor of a and b also divides $\gcd(a, b)$.

- If not, then d has a prime factor that $\gcd(a, b)$ does not.
- This prime factor p divides both a and b.
- So, $p \gcd(a, b)$ would divide both a and b, and is larger than $\gcd(a, b)$, which is impossible.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Bijection Facts

Fact 1: For $f : A \rightarrow B$, if A and B are *finite*, then

- a bijection $A \rightarrow B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why?

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.
- Suppose f is surjective.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.
- Suppose f is surjective. If two inputs are mapped to the same output, then $|\operatorname{range} f| < |B|$, impossible.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.
- Suppose f is surjective. If two inputs are mapped to the same output, then $|\operatorname{range} f| < |B|$, impossible.

This is not true for infinite sets.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.
- Suppose f is surjective. If two inputs are mapped to the same output, then $|\operatorname{range} f| < |B|$, impossible.

This is not true for infinite sets.

Example: $f : \mathbb{N} \to \mathbb{N}$ with $f(x) = x + 1$ is injective.

Bijection Facts

Fact 1: For $f : A \to B$, if A and B are *finite*, then

- a bijection $A \to B$ exists only if $|A| = |B|$;
- injective $\iff$ surjective $\iff$ bijective.

Why? Counting argument.

- Suppose f is injective. Then $|\operatorname{range} f| = |A| = |B|$, but $\operatorname{range} f \subseteq B$. So $\operatorname{range} f = B$.
- Suppose f is surjective. If two inputs are mapped to the same output, then $|\operatorname{range} f| < |B|$, impossible.

This is not true for infinite sets.
Example: $f : \mathbb{N} \to \mathbb{N}$ with $f(x) = x + 1$ is injective. Not surjective.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$;

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f(g(y)) = y \qquad \text{and} \qquad g(f(x)) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.
- Also, $g(f(x)) = g(y) = x$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f(g(y)) = y \qquad \text{and} \qquad g(f(x)) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.
- Also, $g(f(x)) = g(y) = x$.
- If g exists, then for $y \in B$, $y = f(g(y))$, where $g(y) \in A$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f(g(y)) = y \qquad \text{and} \qquad g(f(x)) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.
- Also, $g(f(x)) = g(y) = x$.
- If g exists, then for $y \in B$, $y = f(g(y))$, where $g(y) \in A$. So f is surjective.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.
- Also, $g(f(x)) = g(y) = x$.
- If g exists, then for $y \in B$, $y = f(g(y))$, where $g(y) \in A$. So f is surjective.
- If $f(x) = f(y)$, then (apply g) $x = y$.

Bijections Have Inverse Bijections

Fact 2: f is bijective $\iff$ there exists a two-sided inverse function g.

$$f\big(g(y)\big) = y \qquad \text{and} \qquad g\big(f(x)\big) = x$$

for all $x \in A$, $y \in B$.

- If f is bijective, each $y \in B$ has a $x \in A$ with $f(x) = y$; let $g(y) = x$.
- So, $f(g(y)) = f(x) = y$.
- Also, $g(f(x)) = g(y) = x$.
- If g exists, then for $y \in B$, $y = f(g(y))$, where $g(y) \in A$. So f is surjective.
- If $f(x) = f(y)$, then (apply g) $x = y$. So f is injective. $\square$

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

GCD & Bijectivity

Theorem: The map $f(x) = ax \mod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \mod m$ for some x.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.
- Conversely, if $\gcd(a, m) = 1$, then let $ax_1, ax_2 \in \operatorname{range} f$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.
- Conversely, if $\gcd(a, m) = 1$, then let $ax_1, ax_2 \in \operatorname{range} f$.
- If $ax_1 \equiv ax_2 \bmod m$, then $m \mid a(x_1 - x_2)$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.
- Conversely, if $\gcd(a, m) = 1$, then let $ax_1, ax_2 \in \operatorname{range} f$.
- If $ax_1 \equiv ax_2 \bmod m$, then $m \mid a(x_1 - x_2)$.
- But a and m have no common factors, so $m \mid x_1 - x_2$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.
- Conversely, if $\gcd(a, m) = 1$, then let $ax_1, ax_2 \in \text{range} f$.
- If $ax_1 \equiv ax_2 \bmod m$, then $m \mid a(x_1 - x_2)$.
- But a and m have no common factors, so $m \mid x_1 - x_2$.
- Thus, $x_1 \equiv x_2 \pmod{m}$.

GCD & Bijectivity

Theorem: The map $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Proof.

- If f is bijective, then $ax \equiv 1 \bmod m$ for some x.
- So $m \mid ax - 1$.
- So $\gcd(a, m) \mid ax$ and $\gcd(a, m) \mid ax - 1$, which means $\gcd(a, m) \mid 1$. $\gcd(a, m) = 1$.
- Conversely, if $\gcd(a, m) = 1$, then let $ax_1, ax_2 \in \operatorname{range} f$.
- If $ax_1 \equiv ax_2 \bmod m$, then $m \mid a(x_1 - x_2)$.
- But a and m have no common factors, so $m \mid x_1 - x_2$.
- Thus, $x_1 \equiv x_2 \pmod{m}$. So f is injective (and thus bijective because the sets are finite). $\square$

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

- If $\gcd(a, m) = 1$, then $f(x) = ax \bmod m$ is bijective, so there exists x with $ax \equiv 1 \bmod m$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

- If $\gcd(a, m) = 1$, then $f(x) = ax \bmod m$ is bijective, so there exists x with $ax \equiv 1 \bmod m$.
- The multiplicative inverse is unique because f is bijective.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

- If $\gcd(a, m) = 1$, then $f(x) = ax \bmod m$ is bijective, so there exists x with $ax \equiv 1 \bmod m$.
- The multiplicative inverse is unique because f is bijective.
- On the other hand, if $d := \gcd(a, m) > 1$, then $m/d \not\equiv 0 \pmod{m}$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

- If $\gcd(a, m) = 1$, then $f(x) = ax \bmod m$ is bijective, so there exists x with $ax \equiv 1 \bmod m$.
- The multiplicative inverse is unique because f is bijective.
- On the other hand, if $d := \gcd(a, m) > 1$, then $m/d \not\equiv 0 \pmod{m}$.
- So for any x, $ax \cdot (m/d) \equiv x(a/d) \cdot m \equiv 0 \pmod{m}$.

Existence of Multiplicative Inverses

Theorem: $f(x) = ax \bmod m$ is bijective if and only if $\gcd(a, m) = 1$.

For $a \in \mathbb{Z}/m\mathbb{Z}$, a **multiplicative inverse** x is an element of $\mathbb{Z}/m\mathbb{Z}$ for which $ax \equiv 1 \pmod{m}$.

Corollary: For all $a \in \mathbb{Z}/m\mathbb{Z}$, a has a multiplicative inverse (necessarily unique) if and only if $\gcd(a, m) = 1$.

- If $\gcd(a, m) = 1$, then $f(x) = ax \bmod m$ is bijective, so there exists x with $ax \equiv 1 \bmod m$.
- The multiplicative inverse is unique because f is bijective.
- On the other hand, if $d := \gcd(a, m) > 1$, then $m/d \not\equiv 0 \pmod{m}$.
- So for any x, $ax \cdot (m/d) \equiv x(a/d) \cdot m \equiv 0 \pmod{m}$.
- So no multiplicative inverse for a can exist.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse?

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Notation: $(\mathbb{Z}/m\mathbb{Z})^{\times}$ consists of the elements in $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Notation: $(\mathbb{Z}/m\mathbb{Z})^\times$ consists of the elements in $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses.

- So, $a \in (\mathbb{Z}/m\mathbb{Z})^\times$ if and only if $\gcd(a, m) = 1$.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Notation: $(\mathbb{Z}/m\mathbb{Z})^{\times}$ consists of the elements in $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses.

- So, $a \in (\mathbb{Z}/m\mathbb{Z})^{\times}$ if and only if $\gcd(a, m) = 1$.
- Example: $(\mathbb{Z}/6\mathbb{Z})^{\times} = \{1, 5\}$.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Notation: $(\mathbb{Z}/m\mathbb{Z})^\times$ consists of the elements in $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses.

- So, $a \in (\mathbb{Z}/m\mathbb{Z})^\times$ if and only if $\gcd(a, m) = 1$.
- Example: $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$.
- Example: $(\mathbb{Z}/8\mathbb{Z})^\times = \{1, 3, 5, 7\}$.

Elements with Multiplicative Inverses

If a^{-1} exists in $\mathbb{Z}/m\mathbb{Z}$, then a^{-1} also has an inverse. Namely, a is the inverse of a^{-1}.

Consequence: $\gcd(a^{-1}, m) = 1$.

If a and b have inverses, does ab have an inverse? Yes, $a^{-1}b^{-1}$.

Notation: $(\mathbb{Z}/m\mathbb{Z})^\times$ consists of the elements in $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses.

- So, $a \in (\mathbb{Z}/m\mathbb{Z})^\times$ if and only if $\gcd(a, m) = 1$.
- Example: $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$.
- Example: $(\mathbb{Z}/8\mathbb{Z})^\times = \{1, 3, 5, 7\}$.
- Example: $(\mathbb{Z}/p\mathbb{Z})^\times = \{1, \ldots, p-1\}$ for p prime.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.
- Or, $1 + 1 = 2$ does not have an inverse.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.
- Or, $1 + 1 = 2$ does not have an inverse.

When p is prime, $\mathbb{Z}/p\mathbb{Z}$ is more special: any non-zero number has an inverse.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.
- Or, $1 + 1 = 2$ does not have an inverse.

When p is prime, $\mathbb{Z}/p\mathbb{Z}$ is more special: any non-zero number has an inverse. Like $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.
- Or, $1 + 1 = 2$ does not have an inverse.

When p is prime, $\mathbb{Z}/p\mathbb{Z}$ is more special: any non-zero number has an inverse. Like $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$.

So, $\mathbb{Z}/p\mathbb{Z}$ is called a **field**.

The Structure of $(\mathbb{Z}/m\mathbb{Z})^\times$

In $(\mathbb{Z}/m\mathbb{Z})^\times$, not only can we multiply, we can also divide. Multiplicative inverses exist!

But we can no longer *add*.

- In $(\mathbb{Z}/6\mathbb{Z})^\times = \{1, 5\}$, notice that $1 + 5 = 0$ does not have an inverse.
- Or, $1 + 1 = 2$ does not have an inverse.

When p is prime, $\mathbb{Z}/p\mathbb{Z}$ is more special: any non-zero number has an inverse. Like $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$.

So, $\mathbb{Z}/p\mathbb{Z}$ is called a **field**. Sometimes, this is called GF(p).

Computing the GCD

Given $a, b \in \mathbb{Z}$, how do we calculate $\gcd(a, b)$?

Computing the GCD

Given $a, b \in \mathbb{Z}$, how do we calculate $\gcd(a, b)$?

First approach: factor a and b.

Computing the GCD

Given $a, b \in \mathbb{Z}$, how do we calculate $\gcd(a, b)$?

First approach: factor a and b.

- Example: Let $72 = 2^3 \cdot 3^2$ and $27 = 3^3$.

Computing the GCD

Given $a, b \in \mathbb{Z}$, how do we calculate $\gcd(a, b)$?

First approach: factor a and b.

- Example: Let $72 = 2^3 \cdot 3^2$ and $27 = 3^3$.
- For each prime p, take the largest power of p that divides both numbers.

Computing the GCD

Given $a, b \in \mathbb{Z}$, how do we calculate $\gcd(a, b)$?

First approach: factor a and b.

- Example: Let $72 = 2^3 \cdot 3^2$ and $27 = 3^3$.
- For each prime p, take the largest power of p that divides both numbers.
- Here, the GCD is $3^2 = 9$.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?
- We want an algorithm that runs in time which is a *polynomial* in the size of the input.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?
- We want an algorithm that runs in time which is a *polynomial* in the size of the input.
- For a positive integer N, it takes $\approx \log_2 N$ bits to write.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?
- We want an algorithm that runs in time which is a *polynomial* in the size of the input.
- For a positive integer N, it takes $\approx \log_2 N$ bits to write. We can try dividing N by all numbers between 1 and N.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?
- We want an algorithm that runs in time which is a *polynomial* in the size of the input.
- For a positive integer N, it takes $\approx \log_2 N$ bits to write. We can try dividing N by all numbers between 1 and N.
- The above algorithm runs in time $O(N)$, then its runtime is *exponential* in the input size.

Factoring Is Slow?

Problem: We do not know how to factor numbers *fast*.

- What does *fast* mean?
- We want an algorithm that runs in time which is a *polynomial* in the size of the input.
- For a positive integer N, it takes $\approx \log_2 N$ bits to write. We can try dividing N by all numbers between 1 and N.
- The above algorithm runs in time $O(N)$, then its runtime is *exponential* in the input size.
- Actually we only have to check $O(\sqrt{N})$ numbers, but this is still bad—we want $O((\log_2 N)^k)$ for some $k \in \mathbb{N}$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Euclid's Algorithm

Given positive integers *a*, *b*, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Example: $a = 72$, $b = 27$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Example: $a = 72$, $b = 27$.

- $\gcd(72, 27) = \gcd(27, 72 \bmod 27) = \gcd(27, 18)$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Example: $a = 72$, $b = 27$.

- $\gcd(72, 27) = \gcd(27, 72 \bmod 27) = \gcd(27, 18)$.
- $\gcd(27, 18) = \gcd(18, 27 \bmod 18) = \gcd(18, 9)$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Example: $a = 72$, $b = 27$.

- $\gcd(72, 27) = \gcd(27, 72 \bmod 27) = \gcd(27, 18)$.
- $\gcd(27, 18) = \gcd(18, 27 \bmod 18) = \gcd(18, 9)$.
- $\gcd(18, 9) = \gcd(9, 18 \bmod 9) = \gcd(9, 0)$.

Euclid's Algorithm

Given positive integers a, b, assume (WLOG) $a > b$.

Key observation: If we write $a = qb + r$ (by the Division Algorithm), where $q \in \mathbb{Z}$ and $r \in \{0, 1, \ldots, b-1\}$, then:

- If d is a common divisor of a and b, then $d \mid a - qb = r$.
- If d is a common divisor of b and r, then $d \mid qb + r = a$.
- A number divides a and b if and only if it divides b and r.

In other words, $\gcd(a, b) = \gcd(b, a \bmod b)$.

Example: $a = 72$, $b = 27$.

- $\gcd(72, 27) = \gcd(27, 72 \bmod 27) = \gcd(27, 18)$.
- $\gcd(27, 18) = \gcd(18, 27 \bmod 18) = \gcd(18, 9)$.
- $\gcd(18, 9) = \gcd(9, 18 \bmod 9) = \gcd(9, 0)$.
- $\gcd(9, 0) = 9$.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.
- Case 1: If $b < a/2$, then in one iteration the first argument is cut in half.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.
- Case 1: If $b < a/2$, then in one iteration the first argument is cut in half.
- In two iterations, the first argument becomes $a \bmod b$.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.
- Case 1: If $b < a/2$, then in one iteration the first argument is cut in half.
- In two iterations, the first argument becomes $a \bmod b$.
- Case 2: If $b \geq a/2$, then $a \bmod b \leq a/2$.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a, 0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.
- Case 1: If $b < a/2$, then in one iteration the first argument is cut in half.
- In two iterations, the first argument becomes $a \bmod b$.
- Case 2: If $b \geq a/2$, then $a \bmod b \leq a/2$. The first argument is cut in half.

Analysis of Euclid's Algorithm

Euclid's Algorithm: Given two positive integers $a > b$:

- If $b = 0$, then $\gcd(a,0) = a$.
- Otherwise, set $a := b$ and $b := a \bmod b$.
- Repeat.

Analysis: What happens to the first argument, a?

- In one iteration, the first argument becomes b.
- Case 1: If $b < a/2$, then in one iteration the first argument is cut in half.
- In two iterations, the first argument becomes $a \bmod b$.
- Case 2: If $b \geq a/2$, then $a \bmod b \leq a/2$. The first argument is cut in half.

In at most two iterations, the first argument is cut in half.

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

If a has $\log_2 N$ bits, then it takes $\approx 2 \log_2 N$ iterations to lose all of its bits.

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

If a has $\log_2 N$ bits, then it takes $\approx 2\log_2 N$ iterations to lose all of its bits.

In each iteration, we perform a division, so it takes $O(\log N)$ divisions.

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

If a has $\log_2 N$ bits, then it takes $\approx 2\log_2 N$ iterations to lose all of its bits.

In each iteration, we perform a division, so it takes $O(\log N)$ divisions.

If $a = 2^{100}\ldots$

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, “cut in half” means “lose a bit”.

If a has $\log_2 N$ bits, then it takes $\approx 2\log_2 N$ iterations to lose all of its bits.

In each iteration, we perform a division, so it takes $O(\log N)$ divisions.

If $a = 2^{100}\ldots$

- If we try all numbers from 1 to $\sqrt{a}$, we need to check 2^{50} numbers.

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

If a has $\log_2 N$ bits, then it takes $\approx 2\log_2 N$ iterations to lose all of its bits.

In each iteration, we perform a division, so it takes $O(\log N)$ divisions.

If $a = 2^{100}\ldots$

- If we try all numbers from 1 to $\sqrt{a}$, we need to check 2^{50} numbers. About one quadrillion numbers!

Analysis of Euclid's Algorithm

In at most two iterations, the first argument is cut in half.

In binary, "cut in half" means "lose a bit".

If a has $\log_2 N$ bits, then it takes $\approx 2\log_2 N$ iterations to lose all of its bits.

In each iteration, we perform a division, so it takes $O(\log N)$ divisions.

If $a = 2^{100}\dots$

- If we try all numbers from 1 to $\sqrt{a}$, we need to check 2^{50} numbers. About one quadrillion numbers!
- If we use Euclid, we need ≈ 200 divisions.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.
- So, $my = ax - 1$ for some $y \in \mathbb{Z}$.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.
- So, $my = ax - 1$ for some $y \in \mathbb{Z}$. (definition of divisibility)

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.
- So, $my = ax - 1$ for some $y \in \mathbb{Z}$. (definition of divisibility)
- So, $ax - my = 1$ for some $x, y \in \mathbb{Z}$.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.
- So, $my = ax - 1$ for some $y \in \mathbb{Z}$. (definition of divisibility)
- So, $ax - my = 1$ for some $x, y \in \mathbb{Z}$.

We need to take an integer multiple of a, an integer multiple of m, and add them to form 1.

Looking for Multiplicative Inverses

For $a \in \mathbb{Z}/m\mathbb{Z}$, how do we compute a^{-1} in $\mathbb{Z}/m\mathbb{Z}$?

- The inverse is a number x such that $ax \equiv 1 \pmod{m}$.
- So, $m \mid ax - 1$.
- So, $my = ax - 1$ for some $y \in \mathbb{Z}$. (definition of divisibility)
- So, $ax - my = 1$ for some $x, y \in \mathbb{Z}$.

We need to take an integer multiple of a, an integer multiple of m, and add them to form 1.

Next question to investigate: what numbers can we reach using integer combinations of a and m?

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Second observation: Since this holds for any common divisor, it holds for the *greatest* common divisor $\gcd(a, m)$.

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Second observation: Since this holds for any common divisor, it holds for the *greatest* common divisor $\gcd(a, m)$.

So, the only numbers we can reach are multiples of $\gcd(a, m)$.

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Second observation: Since this holds for any common divisor, it holds for the *greatest* common divisor $\gcd(a, m)$.

So, the only numbers we can reach are multiples of $\gcd(a, m)$.

- This (again) proves that if $\gcd(a, m) \neq 1$, then a^{-1} does not exist in $\mathbb{Z}/m\mathbb{Z}$.

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Second observation: Since this holds for any common divisor, it holds for the *greatest* common divisor $\gcd(a, m)$.

So, the only numbers we can reach are multiples of $\gcd(a, m)$.

- This (again) proves that if $\gcd(a, m) \neq 1$, then a^{-1} does not exist in $\mathbb{Z}/m\mathbb{Z}$.
- Since we can only reach multiples of $\gcd(a, m)$ with integer linear combinations of a and m, then we can never form 1.

Integer Linear Combinations

What numbers can we reach using integer linear combinations of a and m?

First observation: If d divides a and m, then d divides any integer linear combination of a and m.

Second observation: Since this holds for any common divisor, it holds for the *greatest* common divisor $\gcd(a, m)$.

So, the only numbers we can reach are multiples of $\gcd(a, m)$.

- This (again) proves that if $\gcd(a, m) \neq 1$, then a^{-1} does not exist in $\mathbb{Z}/m\mathbb{Z}$.
- Since we can only reach multiples of $\gcd(a, m)$ with integer linear combinations of a and m, then we can never form 1.

Goal: Express $\gcd(a, m)$ as an integer combination of a and m.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a, b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a,b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a,b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a, b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders. The remainder left at the end is the GCD.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a, b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders. The remainder left at the end is the GCD.

Can we write each remainder as an integer combination of a and b?

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a, b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders. The remainder left at the end is the GCD.

Can we write each remainder as an integer combination of a and b?

- Start with $r = 1 \cdot a - q \cdot b$.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a, b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a, b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders. The remainder left at the end is the GCD.

Can we write each remainder as an integer combination of a and b?

- Start with $r = 1 \cdot a - q \cdot b$.
- The next inputs to the GCD algorithm are b and r.

From Euclid to Multiplicative Inverses

Goal: Express $\gcd(a,b)$ as an integer combination of a and b.

Remember: If we are computing $\gcd(a,b)$, then Euclid's Algorithm uses the Division Algorithm: $a = qb + r$.

Algorithm in a nutshell: keep taking remainders. The remainder left at the end is the GCD.

Can we write each remainder as an integer combination of a and b?

- Start with $r = 1 \cdot a - q \cdot b$.
- The next inputs to the GCD algorithm are b and r.
- Since we have already written $r = 1 \cdot a - q \cdot b$, it is enough to express the next remainder in terms of b and r.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$. Write $9 = 1 \cdot 27 - 1 \cdot 18$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$. Write $9 = 1 \cdot 27 - 1 \cdot 18$.
- Plug in for 18, so $9 = -1 \cdot 72 + 3 \cdot 27$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$. Write $9 = 1 \cdot 27 - 1 \cdot 18$.
- Plug in for 18, so $9 = -1 \cdot 72 + 3 \cdot 27$.
- Next step: $\gcd(18, 9)$.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$. Write $9 = 1 \cdot 27 - 1 \cdot 18$.
- Plug in for 18, so $9 = -1 \cdot 72 + 3 \cdot 27$.
- Next step: $\gcd(18, 9)$.
- The GCD is 9.

Extended Euclid's Algorithm in Action

At each step of the algorithm, express the remainder as an integer linear combination of the inputs.

Example: Let $a = 72$, $b = 27$.

- Start with $\gcd(72, 27)$.
- Division Algorithm: $72 = 2 \cdot 27 + 18$. Write $18 = 1 \cdot 72 - 2 \cdot 27$.
- Next step: $\gcd(27, 18)$.
- Division Algorithm: $27 = 1 \cdot 18 + 9$. Write $9 = 1 \cdot 27 - 1 \cdot 18$.
- Plug in for 18, so $9 = -1 \cdot 72 + 3 \cdot 27$.
- Next step: $\gcd(18, 9)$.
- The GCD is 9.

We have expressed $9 = \gcd(72, 27) = -1 \cdot 72 + 3 \cdot 27$.

Expressing the Remainder Operation

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

[1] The $\lfloor \cdot \rfloor$ notation is called the **floor** function and it means "round down".

Expressing the Remainder Operation

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

- To calculate $a \bmod b$, first find the largest multiple of b before you hit a.

[1] The $\lfloor \cdot \rfloor$ notation is called the **floor** function and it means "round down".

Expressing the Remainder Operation

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

- To calculate $a \bmod b$, first find the largest multiple of b before you hit a. This is $\lfloor a/b \rfloor b$.[1]

[1] The $\lfloor \cdot \rfloor$ notation is called the **floor** function and it means "round down".

Expressing the Remainder Operation

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

- To calculate $a \bmod b$, first find the largest multiple of b before you hit a. This is $\lfloor a/b \rfloor b$.[1]
- Thus the remainder is $a - \lfloor a/b \rfloor b$.

[1] The $\lfloor \cdot \rfloor$ notation is called the **floor** function and it means "round down".

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \mathrm{gcd}(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \mathrm{gcd}(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.
- Then, $\mathrm{egcd}(b, a \bmod b) = (d', x', y')$, where $d' = \mathrm{gcd}(b, a \bmod b) = x' \cdot b + y' \cdot (a \bmod b)$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \mathrm{gcd}(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.
- Then, $\mathrm{egcd}(b, a \bmod b) = (d', x', y')$, where $d' = \mathrm{gcd}(b, a \bmod b) = x' \cdot b + y' \cdot (a \bmod b)$.
- Now, $\mathrm{gcd}(a, b) = \mathrm{gcd}(b, a \bmod b)$, so set $d := d'$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.
- Then, $\mathrm{egcd}(b, a \bmod b) = (d', x', y')$, where $d' = \gcd(b, a \bmod b) = x' \cdot b + y' \cdot (a \bmod b)$.
- Now, $\gcd(a, b) = \gcd(b, a \bmod b)$, so set $d := d'$.
- Also, $d = x' \cdot b + y' \cdot (a - \lfloor a/b \rfloor b)$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.
- Then, $\mathrm{egcd}(b, a \bmod b) = (d', x', y')$, where $d' = \gcd(b, a \bmod b) = x' \cdot b + y' \cdot (a \bmod b)$.
- Now, $\gcd(a, b) = \gcd(b, a \bmod b)$, so set $d := d'$.
- Also, $d = x' \cdot b + y' \cdot (a - \lfloor a/b \rfloor b)$.
- Rearrange: $d = y' \cdot a + (x' - \lfloor a/b \rfloor y') \cdot b$.

Extended Euclid's Algorithm

Note: $a \bmod b = a - \lfloor a/b \rfloor b$.

Extended Euclid's Algorithm:

- Goal: Given positive integers $a > b$, return (d, x, y), where $d = \gcd(a, b)$, and $d = x \cdot a + y \cdot b$.
- Base case: If $b = 0$, then $\text{egcd}(a, 0) = (a, 1, 0)$. Because $a = 1 \cdot a + 0 \cdot 0$.
- Assume (strong induction) that extended Euclid works for smaller arguments.
- Then, $\text{egcd}(b, a \bmod b) = (d', x', y')$, where $d' = \gcd(b, a \bmod b) = x' \cdot b + y' \cdot (a \bmod b)$.
- Now, $\gcd(a, b) = \gcd(b, a \bmod b)$, so set $d := d'$.
- Also, $d = x' \cdot b + y' \cdot (a - \lfloor a/b \rfloor b)$.
- Rearrange: $d = y' \cdot a + (x' - \lfloor a/b \rfloor y') \cdot b$.
- So, set $x := y'$ and $y := x' - \lfloor a/b \rfloor y'$.

Extended Euclid's Algorithm

Extended Euclid's Algorithm:

- If $b = 0$, then $\mathrm{egcd}(a, 0) = (a, 1, 0)$.
- Otherwise, let $(d', x', y') := \mathrm{egcd}(b, a \bmod b)$. Return $(d', y', x' - \lfloor a/b \rfloor y')$.

Summary

- We proved facts about bijections.
- The element $a \in \mathbb{Z}/m\mathbb{Z}$ has a multiplicative inverse (i.e., $a \in (\mathbb{Z}/m\mathbb{Z})^{\times}$) if and only if $\gcd(a, m) = 1$.
- Euclid's Algorithm: Efficiently compute GCD.
- Extended Euclid: Efficiently express GCD as an integer linear combination of the inputs.