CSE 311 Lecture 14: Euclidean Algorithm and Modular Equations

Emina Torlak and Sami Davies

Topics

Primes and GCD

A quick review of Lecture 13.

Extended Euclidean algorithm

Bézout's theorem and the extended Euclidean algorithm.

Modular equations

Solving modular equations with the extended Euclidean algorithm.

Primes and GCD

A quick review of Lecture 13.

Primes and composites: definitions and theorems

Prime number

An integer $p > 1$ is called *prime* if its only positive factors are 1 and p.

Composite number

An integer $c > 1$ is called *composite* if it is not prime.

Fundamental theorem of arithmetic

Every positive integer greater than 1 has a unique prime factorization.

Euclid's theorem

There are infinitely many primes.

Greatest common divisor (GCD): definition

Greatest common divisor (GCD)

The greatest common divisor of integers a and b, written as $\mathrm{GCD}(a, b)$, is the largest integer d such that $d \mid a$ and $d \mid b$.

We can compute GCDs efficiently using the Euclidean algorithm. Invented in 300 BC!

Euclidean algorithm: review

Euclidean algorithm is based on two useful facts:

$\mathrm{GCD}(a, 0) = a$ for all positive integers a.

$\mathrm{GCD}(a, b) = \mathrm{GCD}(b, a \bmod b)$ for all positive integers a and b.

Example implementation:

```
// Assumes a >= b >= 0.
public static int gcd(int a, int b) {
  if (b == 0)
    return a;            // GCD(a, 0) = a
  else
    return gcd(b, a % b); // GCD(a, b) = GCD(b, a mod b)
}
```

GCD(660, 126)

$= \mathrm{GCD}(126, 660 \bmod 126) = \mathrm{GCD}(126, 30)$

$= \mathrm{GCD}(30, 126 \bmod 30) = \mathrm{GCD}(30, 6)$

$= \mathrm{GCD}(6, 30 \bmod 6) = \mathrm{GCD}(6, 0)$

$= 6$

In *tableau form*:

660 = 5 * 126 + 30
126 = 4 * 30 + 6
30 = 5 * 6 + 0

Extended Euclidean algorithm

Bézout’s theorem and the extended Euclidean algorithm.

Bézout's theorem about GCDs

Bézout's theorem

If a and b are positive integers, then there exist integers s and t such that $\mathrm{GCD}(a, b) = sa + tb$.

Bézout's theorem about GCDs

Bézout's theorem

If a and b are positive integers, then there exist integers s and t such that $\mathrm{GCD}(a, b) = sa + tb$.

We can extend Euclidean algorithm to find s and t in addition to computing $\mathrm{GCD}(a, b)$.

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$\boxed{\begin{array}{rcrcrcr} a & = & q & * & b & + & r \\ 35 & = & 1 & * & 27 & + & 8 \\ 27 & = & 3 & * & 8 & + & 3 \\ 8 & = & 2 & * & 3 & + & 2 \\ 3 & = & 1 & * & 2 & + & 1 \end{array}}$$

$$\begin{array}{rclcl} \mathrm{GCD}(a, b) & & \mathrm{GCD}(b, a \bmod b) & & r = a \bmod b \\ \mathrm{GCD}(35, 27) & = & \mathrm{GCD}(27, 35 \bmod 27) & = & \mathrm{GCD}(27, 8) \\ & = & \mathrm{GCD}(8, 27 \bmod 8) & = & \mathrm{GCD}(8, 3) \\ & = & \mathrm{GCD}(3, 8 \bmod 3) & = & \mathrm{GCD}(3, 2) \\ & = & \mathrm{GCD}(2, 3 \bmod 2) & = & \mathrm{GCD}(2, 1) \\ & = & \mathrm{GCD}(1, 2 \bmod 1) & = & \mathrm{GCD}(1, 0) \end{array}$$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$\boxed{\begin{array}{rcccccc} a & = & q & * & b & + & r \\ 35 & = & 1 & * & 27 & + & 8 \\ 27 & = & 3 & * & 8 & + & 3 \\ 8 & = & 2 & * & 3 & + & 2 \\ 3 & = & 1 & * & 2 & + & 1 \end{array}}$$

$$\begin{array}{rcccccc} r & = & a & - & q & * & b \\ 8 & = & 35 & - & 1 & * & 27 \\ 3 & = & 27 & - & 3 & * & 8 \\ 2 & = & 8 & - & 2 & * & 3 \\ 1 & = & 3 & - & 1 & * & 2 \end{array}$$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$
\begin{array}{ccrcccr}
r & = & a & - & q & * & b \\
8 & = & 35 & - & 1 & * & 27 \\
3 & = & 27 & - & 3 & * & 8 \\
2 & = & 8 & - & 2 & * & 3 \\
1 & = & 3 & - & 1 & * & 2
\end{array}
$$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$
\begin{array}{rcrcrcr}
r & = & a & - & q & * & b \\
8 & = & 35 & - & 1 & * & 27 \\
3 & = & 27 & - & 3 & * & 8 \\
2 & = & 8 & - & 2 & * & 3 \\
1 & = & 3 & - & 1 & * & 2
\end{array}
$$

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$
\begin{array}{ccccccc}
r & = & a & - & q & * & b \\
8 & = & 35 & - & 1 & * & 27 \\
3 & = & 27 & - & 3 & * & 8 \\
2 & = & 8 & - & 2 & * & 3 \\
1 & = & 3 & - & 1 & * & 2
\end{array}
$$

$$r_i = r_{i-2} - q_i * r_{i-1}$$
$$r_0 = a = 35$$
$$r_1 = b = 27$$
$$r_2 = r_0 - q_2 * r_1 = 8$$
$$r_3 = r_1 - q_3 * r_2 = 3$$
$$r_4 = r_2 - q_4 * r_3 = 2$$
$$r_5 = r_3 - q_5 * r_4 = 1$$

$$1 = 3 - 1 * 2$$

$$r_5 = r_3 - q_5 * r_4.$$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$\begin{array}{ccccccc} r & = & a & - & q & * & b \\ 8 & = & 35 & - & 1 & * & 27 \\ 3 & = & 27 & - & 3 & * & 8 \\ 2 & = & 8 & - & 2 & * & 3 \\ 1 & = & 3 & - & 1 & * & 2 \end{array}$$

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

$$\begin{aligned} 1 &= 3 - 1 * 2 \\ &= 3 - 1 * (8 - 2 * 3) \end{aligned}$$

$r_5 = r_3 - q_5 * r_4.$

Plug in $r_4 = r_2 - q_4 * r_3.$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$\begin{array}{ccccccc} r & = & a & - & q & * & b \\ 8 & = & 35 & - & 1 & * & 27 \\ 3 & = & 27 & - & 3 & * & 8 \\ 2 & = & 8 & - & 2 & * & 3 \\ 1 & = & 3 & - & 1 & * & 2 \end{array}$$

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

$$\begin{aligned} 1 &= 3 - 1 * 2 \\ &= 3 - 1 * (8 - 2 * 3) \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

$r_5 = r_3 - q_5 * r_4.$

Plug in $r_4 = r_2 - q_4 * r_3.$

Combine r_2, r_3 terms.

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$
\begin{array}{ccccccc}
r & = & a & - & q & * & b \\
8 & = & 35 & - & 1 & * & 27 \\
3 & = & 27 & - & 3 & * & 8 \\
2 & = & 8 & - & 2 & * & 3 \\
1 & = & 3 & - & 1 & * & 2
\end{array}
$$

$$r_i = r_{i-2} - q_i * r_{i-1}$$
$$r_0 = a = 35$$
$$r_1 = b = 27$$
$$r_2 = r_0 - q_2 * r_1 = 8$$
$$r_3 = r_1 - q_3 * r_2 = 3$$
$$r_4 = r_2 - q_4 * r_3 = 2$$
$$r_5 = r_3 - q_5 * r_4 = 1$$

$$
\begin{aligned}
1 &= 3 - 1 * 2 \\
&= 3 - 1 * (8 - 2 * 3) \\
&= (-1) * 8 + 3 * 3 \\
&= (-1) * 8 + 3 * (27 - 3 * 8)
\end{aligned}
$$

$r_5 = r_3 - q_5 * r_4$.

Plug in $r_4 = r_2 - q_4 * r_3$.

Combine r_2, r_3 terms.

Plug in $r_3 = r_1 - q_3 * r_2$.

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$\begin{array}{ccccccc} r & = & a & - & q & * & b \\ 8 & = & 35 & - & 1 & * & 27 \\ 3 & = & 27 & - & 3 & * & 8 \\ 2 & = & 8 & - & 2 & * & 3 \\ 1 & = & 3 & - & 1 & * & 2 \end{array}$$

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

$$\begin{aligned} 1 &= 3 - 1 * 2 \\ &= 3 - 1 * (8 - 2 * 3) \\ &= (-1) * 8 + 3 * 3 \\ &= (-1) * 8 + 3 * (27 - 3 * 8) \\ &= 3 * 27 + (-10) * 8 \end{aligned}$$

$r_5 = r_3 - q_5 * r_4$.

Plug in $r_4 = r_2 - q_4 * r_3$.

Combine r_2, r_3 terms.

Plug in $r_3 = r_1 - q_3 * r_2$.

Combine r_1, r_2 terms.

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

$$
\begin{array}{ccccccc}
r & = & a & - & q & * & b \\
8 & = & 35 & - & 1 & * & 27 \\
3 & = & 27 & - & 3 & * & 8 \\
2 & = & 8 & - & 2 & * & 3 \\
1 & = & 3 & - & 1 & * & 2
\end{array}
$$

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

$1 = 3 - 1 * 2$	$r_5 = r_3 - q_5 * r_4.$
$= 3 - 1 * (8 - 2 * 3)$	Plug in $r_4 = r_2 - q_4 * r_3.$
$= (-1) * 8 + 3 * 3$	Combine r_2, r_3 terms.
$= (-1) * 8 + 3 * (27 - 3 * 8)$	Plug in $r_3 = r_1 - q_3 * r_2.$
$= 3 * 27 + (-10) * 8$	Combine r_1, r_2 terms.
$= 3 * 27 + (-10) * (35 - 1 * 27)$	Plug in $r_2 = r_0 - q_2 * r_1.$

Extended Euclidean algorithm

1. Compute GCD and keep the tableau.
2. Solve the equations for r in the tableau.
3. Back substitute the equations for r.

$\mathrm{GCD}(35, 27) = 35s + 27t.$

r	$=$	a	$-$	q	$*$	b
8	=	35	−	1	*	27
3	=	27	−	3	*	8
2	=	8	−	2	*	3
1	=	3	−	1	*	2

$r_i = r_{i-2} - q_i * r_{i-1}$

$r_0 = a = 35$

$r_1 = b = 27$

$r_2 = r_0 - q_2 * r_1 = 8$

$r_3 = r_1 - q_3 * r_2 = 3$

$r_4 = r_2 - q_4 * r_3 = 2$

$r_5 = r_3 - q_5 * r_4 = 1$

$1 = 3 - 1 * 2$	$r_5 = r_3 - q_5 * r_4.$
$= 3 - 1 * (8 - 2 * 3)$	Plug in $r_4 = r_2 - q_4 * r_3.$
$= (-1) * 8 + 3 * 3$	Combine r_2, r_3 terms.
$= (-1) * 8 + 3 * (27 - 3 * 8)$	Plug in $r_3 = r_1 - q_3 * r_2.$
$= 3 * 27 + (-10) * 8$	Combine r_1, r_2 terms.
$= 3 * 27 + (-10) * (35 - 1 * 27)$	Plug in $r_2 = r_0 - q_2 * r_1.$
$= (-10) * 35 + 13 * 27$	Combine r_0, r_1 terms.

Multiplicative inverse $\mathbf{mod}\ \boldsymbol{m}$

Suppose $\mathrm{GCD}(a, m) = 1$.

Multiplicative inverse $\bmod m$

Suppose $\mathrm{GCD}(a, m) = 1$.

By Bézout's theorem, there exist integers s and t such that $sa + tm = 1$.

Multiplicative inverse $\bmod m$

Suppose $\mathrm{GCD}(a, m) = 1$.

By Bézout's theorem, there exist integers s and t such that $sa + tm = 1$.

$s \bmod m$ is the *multiplicative inverse* of a modulo m: $(s \bmod m)a \equiv 1 \pmod m$

To see why, note that $sa \equiv 1 \pmod m$ and $s \equiv s \bmod m \pmod m$, so by the multiplication property, $(s \bmod m)a \equiv sa \pmod m$, and by transitivity of congruence modulo m, we have that $(s \bmod m)a \equiv 1 \pmod m$.

Multiplicative inverse $\bmod m$

Suppose $\mathrm{GCD}(a, m) = 1$.

By Bézout's theorem, there exist integers s and t such that $sa + tm = 1$.

$s \bmod m$ is the *multiplicative inverse* of a modulo m: $(s \bmod m)a \equiv 1 \pmod{m}$

To see why, note that $sa \equiv 1 \pmod{m}$ and $s \equiv s \bmod m \pmod{m}$, so by the multiplication property, $(s \bmod m)a \equiv sa \pmod{m}$, and by transitivity of congruence modulo m, we have that $(s \bmod m)a \equiv 1 \pmod{m}$.

So, we can compute multiplicative inverses with the extended Euclidean algorithm. These inverses let us solve modular equations.

Modular equations

Solving modular equations with the extended Euclidean algorithm.

Using multiplicative inverses to solve modular equations

Solve: $7x \equiv 1 \pmod{26}$

Using multiplicative inverses to solve modular equations

Solve: $7x \equiv 1 \pmod{26}$

① Compute GCD and keep the tableau.

$$\begin{aligned} \text{GCD}(26, 7) &= \text{GCD}(7, 5) = \text{GCD}(5, 2) \\ &= \text{GCD}(2, 1) = \text{GCD}(1, 0) \\ &= 1 \end{aligned}$$

Using multiplicative inverses to solve modular equations

Solve: $7x \equiv 1 \pmod{26}$

① Compute GCD and keep the tableau.

$$\begin{aligned} \text{GCD}(26, 7) &= \text{GCD}(7, 5) = \text{GCD}(5, 2) \\ &= \text{GCD}(2, 1) = \text{GCD}(1, 0) \\ &= 1 \end{aligned}$$

② Solve the equations for r in the tableau.

$$\boxed{\begin{array}{ccccccccc} a & = & q & * & b & + & r \\ 26 & = & 3 & * & 7 & + & 5 \\ 7 & = & 1 & * & 5 & + & 2 \\ 5 & = & 2 & * & 2 & + & 1 \end{array}} \qquad \begin{array}{ccccccc} r & = & a & - & q & * & b \\ 5 & = & 26 & - & 3 & * & 7 \\ 2 & = & 7 & - & 1 & * & 5 \\ 1 & = & 5 & - & 2 & * & 2 \end{array}$$

Using multiplicative inverses to solve modular equations

Solve: $7x \equiv 1 \pmod{26}$

① Compute GCD and keep the tableau.

$$\begin{aligned} \mathrm{GCD}(26, 7) &= \mathrm{GCD}(7, 5) = \mathrm{GCD}(5, 2) \\ &= \mathrm{GCD}(2, 1) = \mathrm{GCD}(1, 0) \\ &= 1 \end{aligned}$$

② Solve the equations for r in the tableau.

$$\boxed{\begin{array}{ccccccccc} a & = & q & * & b & + & r \\ 26 & = & 3 & * & 7 & + & 5 \\ 7 & = & 1 & * & 5 & + & 2 \\ 5 & = & 2 & * & 2 & + & 1 \end{array}}$$

$$\begin{array}{ccccccc} r & = & a & - & q & * & b \\ 5 & = & 26 & - & 3 & * & 7 \\ 2 & = & 7 & - & 1 & * & 5 \\ 1 & = & 5 & - & 2 & * & 2 \end{array}$$

③ Back substitute the equations for r.

$$\begin{aligned} 1 &= 5 - 2 * (7 - 1 * 5) \\ &= (-2) * 7 + 3 * 5 \\ &= (-2) * 7 + 3 * (26 - 3 * 7) \\ &= 3 * 26 + (-11) * 7 \end{aligned}$$

Using multiplicative inverses to solve modular equations

Solve: $7x \equiv 1 \pmod{26}$

① Compute GCD and keep the tableau.

$$\begin{aligned} \mathrm{GCD}(26,7) &= \mathrm{GCD}(7,5) = \mathrm{GCD}(5,2) \\ &= \mathrm{GCD}(2,1) = \mathrm{GCD}(1,0) \\ &= 1 \end{aligned}$$

② Solve the equations for r in the tableau.

$$\boxed{\begin{array}{ccccccccc} a & = & q & * & b & + & r \\ 26 & = & 3 & * & 7 & + & 5 \\ 7 & = & 1 & * & 5 & + & 2 \\ 5 & = & 2 & * & 2 & + & 1 \end{array}}$$

$$\begin{array}{ccccccc} r & = & a & - & q & * & b \\ 5 & = & 26 & - & 3 & * & 7 \\ 2 & = & 7 & - & 1 & * & 5 \\ 1 & = & 5 & - & 2 & * & 2 \end{array}$$

③ Back substitute the equations for r.

$$\begin{aligned} 1 &= 5 - 2 * (7 - 1 * 5) \\ &= (-2) * 7 + 3 * 5 \\ &= (-2) * 7 + 3 * (26 - 3 * 7) \\ &= 3 * 26 + (-11) * 7 \end{aligned}$$

④ Solve for x.

- Multiplicative inverse of 7 mod 26
  - $(-11) \bmod 26 = 15$
- So, $x = 26k + 15$ for $k \in Z$.

Solving a more general equation

Solve: $7y \equiv 3 \pmod{26}$

Solving a more general equation

Solve: $7y \equiv 3 \pmod{26}$

We computed that 15 is the multiplicative inverse of 7 modulo 26:

That is, $7 * 15 \equiv 1 \pmod{26}$.

Solving a more general equation

Solve: $7y \equiv 3 \pmod{26}$

We computed that 15 is the multiplicative inverse of 7 modulo 26:

That is, $7 * 15 \equiv 1 \pmod{26}$.

By the multiplication property of mod, we have

$7 * 15 * 3 \equiv 1 * 3 \pmod{26}$.

Solving a more general equation

Solve: $7y \equiv 3 \pmod{26}$

We computed that 15 is the multiplicative inverse of 7 modulo 26:

That is, $7 * 15 \equiv 1 \pmod{26}$.

By the multiplication property of mod, we have

$7 * 15 * 3 \equiv 1 * 3 \pmod{26}$.

So, any $\boldsymbol{y \equiv 15 * 3 \pmod{26}}$ is a solution.

That is, $y = 19 + 26k$ for any $k \in Z$ is a solution.

Solving equations modulo a prime number

$\mathrm{GCD}(a, m) = 1$ if m is prime and $0 < a < m$, so we can always solve modular equations for prime m.

$$a +_7 b = (a + b) \bmod 7$$

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$$a *_7 b = (a * b) \bmod 7$$

*	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

A useful proof technique based on modular equations

Suppose that $x, y \in \mathbb{Z}$ and (x, y) satisfies linear equations

$ax + by = c$ and $dx + ey = f$,

where a, b, c, d, e, f are integer coefficients.

Then (x, y) also satisfies the corresponding equations mod $m > 0 \in \mathbb{Z}$:

$ax + by \equiv c \pmod{m}$ and $dx + ey \equiv f \pmod{m}$.

A useful proof technique based on modular equations

Suppose that $x, y \in \mathbf{Z}$ and (x, y) satisfies linear equations

$ax + by = c$ and $dx + ey = f$,

where a, b, c, d, e, f are integer coefficients.

Then (x, y) also satisfies the corresponding equations mod $m > 0 \in \mathbf{Z}$:

$ax + by \equiv c \pmod{m}$ and $dx + ey \equiv f \pmod{m}$.

The reverse doesn't hold. Can you think of a counterexample?

A useful proof technique based on modular equations

Suppose that $x, y \in \mathbf{Z}$ and (x, y) satisfies linear equations

$ax + by = c$ and $dx + ey = f$,

where a, b, c, d, e, f are integer coefficients.

Then (x, y) also satisfies the corresponding equations mod $m > 0 \in \mathbf{Z}$:

$ax + by \equiv c \pmod{m}$ and $dx + ey \equiv f \pmod{m}$.

The reverse doesn't hold. Can you think of a counterexample?

$(0, 0)$ is a solution to $x + y \equiv 2 \pmod{2}$ and $2x + 2y \equiv 4 \pmod{2}$.

But it's not a solution to $x + y = 2$ and $2x + 2y = 4$.

A useful proof technique based on modular equations

Suppose that $x, y \in \mathbf{Z}$ and (x, y) satisfies linear equations

$ax + by = c$ and $dx + ey = f$,

where a, b, c, d, e, f are integer coefficients.

Then (x, y) also satisfies the corresponding equations mod $m > 0 \in \mathbf{Z}$:

$ax + by \equiv c \pmod{m}$ and $dx + ey \equiv f \pmod{m}$.

The reverse doesn't hold. Can you think of a counterexample?

$(0, 0)$ is a solution to $x + y \equiv 2 \pmod{2}$ and $2x + 2y \equiv 4 \pmod{2}$.

But it's not a solution to $x + y = 2$ and $2x + 2y = 4$.

The contrapositive is a useful proof technique:

You can prove that a system of linear equations with integer coefficients has *no integer solutions* by showing that those equations modulo m have no solutions.

Summary

$\mathbf{GCD}(\boldsymbol{a}, \boldsymbol{b})$ is the greatest integer that divides both $\boldsymbol{a}$ and $\boldsymbol{b}$.

It can be computed efficiently using the Euclidean algorithm.

By Bézout's theorem, $\mathbf{GCD}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{sa} + \boldsymbol{tb}$ for some integers $\boldsymbol{s}, \boldsymbol{t}$.

s, t can be computed using the extended Euclidean algorithm.

If $\mathrm{GCD}(a, b) = 1$, $s \bmod b$ is the multiplicative inverse of a modulo b.

Multiplicative inverses can be used to solve modular equations.