01204211 Discrete Mathematics
Lecture 8b: Modular arithmetic

Jittat Fakcharoenphol

September 25, 2025

Quick check 1

If $a|m$ and $b|m$, can we say that $ab|m$? Prove this fact or provide a counter example.

Quick check 2

If $a|m$, $b|m$, and $a \neq b$ are both prime, can we say that $ab|m$? Prove this fact or provide a counter example.

Prime factorization

One useful fact that we use over and over again is the following.

Unique Factorization (or Fundamental Theorem of Arithmetic)

Every integer greater than 1 can be written *uniquely* as a product of prime numbers (up to the order of factors).

Examples:

- $10 = 2 \cdot 5$
- $13 = 13$
- $112 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 7 = 2^4 \cdot 7$

Problem size

- Sort n numbers

Problem size

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges

Problem size

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices

Problem size

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices
- Add two integers

Problem size

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices
- Add two integers
- Check if an integer n is prime

Problem size

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices
- Add two integers
- Check if an integer n is prime
- Find $gcd(a, b)$ for inputs a and b

Polynomial times

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices

Polynomial times

- Sort n numbers
- Check if vertex u is reachable from vertex v in a graph with n vertices and m edges
- Multiply two $n \times n$ matrices
- Add two integers $O(\log x + \log y)$
- Check if an integer n is prime
- Find $gcd(a, b)$ for inputs a and b

When inputs contain a few numbers

GCD and Power

Days

What day is it today?

Days

What day is it today? Thursday.

Days

What day is it today? ~~Thursday.~~ Monday
What day is 3 days after today?

Days

What day is it today? ~~Thursday.~~ Mon

What day is 3 days after today? ~~Sunday.~~ Thu

Days

What day is it today? ~~Thursday.~~ Mon

What day is 3 days after today? ~~Sunday.~~ Thu

What day is 20 days after today?

Days

What day is it today? Thursday.
What day is 3 days after today? Sunday.
What day is 20 days after today? Wednesday.

Days

What day is it today? Thursday.
What day is 3 days after today? Sunday.
What day is 20 days after today? Wednesday.
What day is 10 days before today?

Days

What day is it today? Thursday.
What day is 3 days after today? Sunday.
What day is 20 days after today? Wednesday.
What day is 10 days before today? Monday.

Clocks

Suppose that it is 1 o'clock.

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours?

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours? 6 o'clock.

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours? 6 o'clock.
What time is the next 10 hours?

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours? 6 o'clock.
What time is the next 10 hours? 11 o'clock.

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours? 6 o'clock.
What time is the next 10 hours? 11 o'clock.
What time is the next 20 hours?

Clocks

Suppose that it is 1 o'clock.
What time is the next 5 hours? 6 o'clock.
What time is the next 10 hours? 11 o'clock.
What time is the next 20 hours? 9 o'clock.

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.

Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 =$$

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.

Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 = 12 \bmod m =$$

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.

Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 = 12 \bmod m = 5.$$

Or

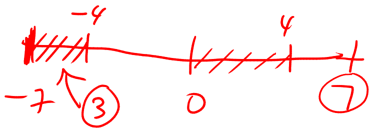
$$2 - 6 =$$

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

$4 \bmod 7 =$

We can then define all arithmetic operations **modulo** m.



Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 = 12 \bmod m = 5.$$

Or

$$2 - 6 = -4 \bmod 7 =$$

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.

Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 = 12 \bmod m = 5.$$

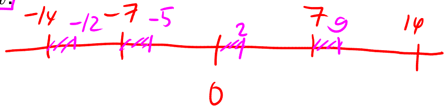
Or

$$2 - 6 = -4 \bmod 7 = 3 \bmod 7 = 3.$$

Modular arithmetic

As in the days of weeks and clocks examples (and also as the modulo in RSA algorithm in our experiment), when working under modular arithmetic, we start with a **modulus** m.

We can then define all arithmetic operations **modulo** m.



Suppose that $m = 7$. We would like to say that

$$4 + 5 = 9 \bmod m = 2.$$

Or

$$3 \cdot 4 = 12 \bmod m = 5.$$

Or

$$2 - 6 = -4 \bmod 7 = 3 \bmod 7 = 3.$$

Note that when you view integers under the lense of modulus 7, these numbers

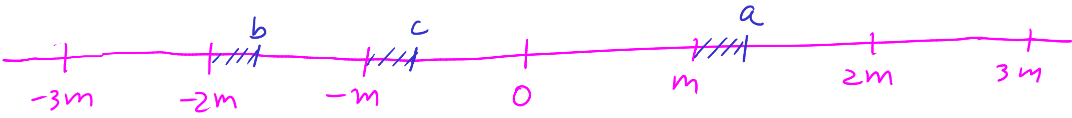
$$\ldots, -19, -12, -5, 2, 9, 16, 23, \ldots$$

are essentially **the same**.

Properties (1)

c mod m = a m

$a \bmod m = b \bmod m$, if and only if $m | a - b$.



Properties (1)

$a \bmod m = b \bmod m$, if and only if $m | a - b$.

Proof.

($\Rightarrow$) Let $r = a \bmod m$. We can write

$$a = qm + r,$$

and

$$b = pm + r,$$

for some integers q and p. Thus, we have

$$a - b = qm + r - pm - r = (q - p)m.$$

Therefore $m | a - b$.

($\Leftarrow$) Exercise. $\square$

Properties (2)

- $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$
- $(a - b) \bmod m = ((a \bmod m) - (b \bmod m)) \bmod m$
- $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$

Congruences

Definition (congruences)

For an integer $m > 0$, if integers a and b are such that

$$a \bmod m = b \bmod m,$$

we write

$$a \equiv b \pmod{m}.$$

Congruences

Definition (congruences)

For an integer $m > 0$, if integers a and b are such that

$$a \bmod m = b \bmod m,$$

we write

$$a \equiv b \pmod{m}.$$

We also have that

$$a \equiv b \pmod{m} \quad \Leftrightarrow \quad m | (a - b)$$

Congruences: properties (1)

$x + 7 \equiv 5 \pmod{13}$

$x + 7 - 7 \equiv 5 - 7 \equiv -2 \equiv 11 \pmod{13}$

$x \equiv 11 \pmod{13}$

- (reflexivity)
  $a \equiv a \pmod{m}$.
- (symmetry)
  $a \equiv b \pmod{m}$ implies $b \equiv a \pmod{m}$.
- (transitiviey)
  $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$ implies $a \equiv c \pmod{m}$.

Congruences: properties (2) – operations

If we have that

$$a \equiv b \pmod{m},$$

and

$$c \equiv d \pmod{m},$$

then

- $a + c \equiv b + d \pmod{m}$
- $a - c \equiv b - d \pmod{m}$
- $ac \equiv bd \pmod{m}$

$2x \equiv 7 \pmod{13}$

$7 \equiv 7 \pmod{13}$

$x \equiv 1 \cdot x \equiv 14x \equiv 7 \cdot 2x \equiv 7 \cdot 7 \equiv 49$

$\equiv 10 \pmod{13}$

$7 \cdot 2 \equiv 1 \pmod{13}$

13
26
39

Congruences: properties (2) – operations

If we have that

$$a \equiv b \pmod{m},$$

and

$$c \equiv d \pmod{m},$$

then

- $a + c \equiv b + d \pmod{m}$
- $a - c \equiv b - d \pmod{m}$
- $ac \equiv bd \pmod{m}$

We can pretty much think of this "congruence" as a normal equation.

Congruences: properties (2) – operations

If we have that

$$a \equiv b \pmod{m},$$

and

$$c \equiv d \pmod{m},$$

then

- $a + c \equiv b + d \pmod{m}$
- $a - c \equiv b - d \pmod{m}$
- $ac \equiv bd \pmod{m}$

We can pretty much think of this "congruence" as a normal equation.

What is missing here?

Congruences: properties (2) – operations

If we have that

$$a \equiv b \pmod{m},$$

and

$$c \equiv d \pmod{m},$$

then

- $a + c \equiv b + d \pmod{m}$
- $a - c \equiv b - d \pmod{m}$
- $ac \equiv bd \pmod{m}$

We can pretty much think of this "congruence" as a normal equation.

What is missing here?
Division!

Also, we wish we can do "cancellation", i.e., if

$$xa \equiv xb \pmod{m},$$

then $a \equiv b \pmod{m}$. **BUT THIS IS NOT ALWAYS TRUE.**

Also, we wish we can do “cancellation”, i.e., if

$$xa \equiv xb \pmod{m},$$

then $a \equiv b \pmod{m}$. **BUT THIS IS NOT ALWAYS TRUE.**
Let’s see the following example:

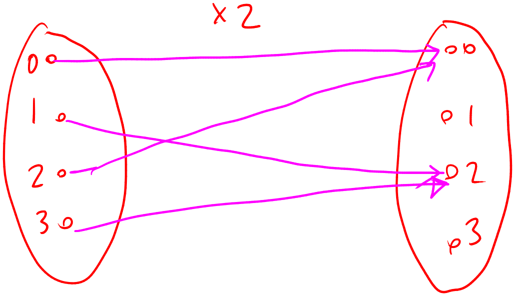
$$2 \cdot 1 \equiv 2 \cdot 3 \pmod{4},$$

but

$$1 \not\equiv 3 \pmod{4}.$$

Multiplications as functions

Let's view multiplication by 2 as a function, i.e., let $f(x) = 2 \cdot x \bmod 4$.

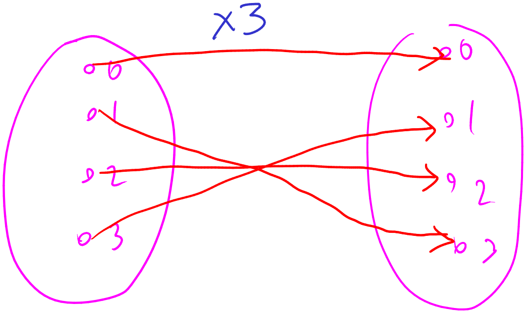


Multiplications as functions

Let's view multiplication by 2 as a function, i.e., let $f(x) = 2 \cdot x \bmod 4$.

Let's also see $g(x) = 3 \cdot x \bmod 4$.

$$3 \cdot x \equiv 2 \pmod 4$$



Multiplications as functions

Let's view multiplication by 2 as a function, i.e., let $f(x) = 2 \cdot x \bmod 4$.

Let's also see $g(x) = 3 \cdot x \bmod 4$.

Which functions have inverses?

Multiplicative inverses (standard arithmetic)

In standard arithmetic, what is $\underline{2/5}$?

$x = 2/5$

Solving

$$\boxed{5x = 2}$$

Multiplicative inverses (standard arithmetic)

In standard arithmetic, what is $2/5$?
We are looking to a number x such that $2 = 5x$. How can we do that?

Multiplicative inverses (standard arithmetic)

In standard arithmetic, what is $2/5$?
We are looking to a number x such that $2 = 5x$. How can we do that?
By dividing on both sides with 5:

$$2/5 = 5x/5 = x,$$

Multiplicative inverses (standard arithmetic)

$$\boxed{\tfrac{2}{5}} = 2\left(\tfrac{1}{5}\right) = 5x\left(\tfrac{1}{5}\right) = \cancel{5}\left(\tfrac{1}{\cancel{5}}\right)x$$

In standard arithmetic, what is 2/5?
We are looking to a number x such that $2 = 5x$. How can we do that?
By dividing on both sides with 5:

$$2/5 = 5x/5 = x,$$

or equivalently, by multiplying with $(1/5) = 5^{-1}$:

$$2 \cdot 5^{-1} = 5x \cdot 5^{-1} = x \cdot (5) \cdot (5^{-1}) = x \cdot \underline{1} = x.$$

Here 5^{-1} is a multiplicative inverse of 5.

Multiplicative inverses (modular arithmetic)

You can do the same thing in modular arithmetic. Let the modulus be $m = 7$. Note that

$$5 \cdot 3 \equiv 15 \equiv 1 \pmod 7.$$

Therefore, $\boxed{5^{-1} \equiv 3 \pmod 7.}$

$$x \equiv 15x \equiv 3 \cdot 5x \equiv \underbrace{2 \cdot 3}_{6} \pmod 7$$

Multiplicative inverses (modular arithmetic)

You can do the same thing in modular arithmetic. Let the modulus be $m = 7$. Note that

$$5 \cdot 3 \equiv 15 \equiv 1 \pmod{7}.$$

Therefore, $5^{-1} \equiv 3 \pmod{7}$.

To find $2/5$, we can view our goal as to find the value of x such that

$$2 \equiv 5x \pmod{7}.$$

We can multiply both sides with $5^{-1} \equiv 3$ to get

$$2 \cdot 5^{-1} \equiv 2 \cdot 3 \equiv 6 \equiv 5^{-1} \cdot 5x \equiv x \pmod{7}.$$

Multiplicative inverses (modular arithmetic)

You can do the same thing in modular arithmetic. Let the modulus be $m = 7$. Note that

$$5 \cdot 3 \equiv 15 \equiv 1 \pmod{7}.$$

Therefore, $5^{-1} \equiv 3 \pmod{7}$.

To find $2/5$, we can view our goal as to find the value of x such that

$$2 \equiv 5x \pmod{7}.$$

We can multiply both sides with $5^{-1} \equiv 3$ to get

$$2 \cdot 5^{-1} \equiv 2 \cdot 3 \equiv 6 \equiv 5^{-1} \cdot 5x \equiv x \pmod{7}.$$

Let's check:

$$5 \cdot 6 \equiv 30 \equiv 2 \pmod{7},$$

as requied.

Multiplicative inverse modulo m

Definition

The multiplicative inverse modulo m of a, denoted by a^{-1}, is an integer such that

$$a \cdot a^{-1} \equiv 1 \pmod{m}.$$

Multiplicative inverse modulo ~~11~~ 13

45
39

Let's try to figure out multiplicative inverse of every integer modulo 1~~1~~3.

a	$a^{-1} \pmod{\cancel{11}}$ 13
1	1
2	7
3	9
4	10
5	8
6	11
7	2
8	5
9	3
10	4
11	6
12	12

27

(1) $\quad 2x + 5y \equiv 6 \pmod{13}$

(2) $\quad x + y \equiv 7 \pmod{13}$

(1) − 2(2) $\quad 3y \equiv 6 - 14$

$\equiv 6 - 1 \equiv 5 \pmod{13}$

$9 \cdot 3y \equiv 27y \equiv 1 \cdot y \equiv 9 \cdot 5$

$\equiv 45 \equiv \boxed{6} \pmod{13}$

$x + 6 \equiv 7$

$\boxed{x \equiv 1}$ $\quad \boxed{y \equiv 6}$ $\quad \pmod{13}$

Multiplicative inverse modulo 11

56

Let's try to figure out multiplicative inverse of every integer modulo 11.

a	$a^{-1} \pmod{11}$
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

$$8(5x + 7y) \equiv 8(7) \pmod{13}$$

$$\boxed{x + 4y \equiv 4 \pmod{13}}$$

Examples: division in modular arithmetic

Suppose that we know that every non-zero integer a has an inverse modulo m. Can you solve this equation?

$$4x + 9 \equiv 0 \pmod{11}.$$

Examples: division in modular arithmetic

Suppose that we know that every non-zero integer a has an inverse modulo m. Can you solve this equation?

$$4x + 9 \equiv 0 \pmod{11}.$$

We can even perform gaussian elimination (*which is very useful later*):

$$\begin{aligned} 2x + y &\equiv 3 \pmod 7 \\ x + 3y &\equiv 5 \pmod 7 \end{aligned}$$

There are 3 clocks. At this moment, all three clocks ring at the same time. The first clock rings every 3 hours, the second clock rings every 4 hours, and the third clock rings every 10 hours. How long do you have to wait until you would hear all clocks ring a the same time again?

You have a large water container and two smaller buckets. The first bucket carries 3 litres of water and the second bucket carries 5 litres of water.
Can you put exactly 1 litre of water in the water container?

You have a large water container and two smaller buckets. The first bucket carries 6 litres of water and the second bucket carries 15 litres of water.
What is the minimum volume of water you can exactly put in the water container?

You have a large water container and two smaller buckets. The first bucket carries 6 litres of water and the second bucket carries 15 litres of water.
What is the minimum volume of water you can exactly put in the water container?

In general if you have two buckets of volumes x and y, the amount that you can exactly make must be in the form of

$$ax + by,$$

for some integers x and y. (Note that x and y may be negative.)

You have a large water container and two smaller buckets. The first bucket carries 6 litres of water and the second bucket carries 15 litres of water.
What is the minimum volume of water you can exactly put in the water container?

In general if you have two buckets of volumes x and y, the amount that you can exactly make must be in the form of

$$ax + by,$$

for some integers x and y. (Note that x and y may be negative.)
Do you see why the sum must be divisible by any common divisor of x and y?

Useful fact

For any integer x and y, consider the term

$$a \cdot x + b \cdot y,$$

for some integer a and b.

Useful fact

For any integer x and y, consider the term

$$a \cdot x + b \cdot y,$$

for some integer a and b.
When the term is non-zero, it must be divisible by $gcd(x, y)$, so it has to be at least $gcd(x, y)$.

It turns out that you can actually attain that value, i.e., there exist a pair of integer a and b such that

$$a \cdot x + b \cdot y = gcd(x, y).$$

Finding a and b: Extended Euclid Algorithm

We will modify the Euclid algorithm so that it also returns a and b together with $gcd(x, y)$.

```
Algorithm Euclid(x,y):
  if x mod y == 0:

    return y,          ,
  else:
    g, a', b' = Euclid(y, x mod y)

    a =

    b =

    return g, a, b
```

Notes:

We have a' and b' such that

$$a' \cdot y + b' \cdot (x \bmod y) = g.$$

Theorem 1

An integer a has a multiplicative inverse modulo m iff $gcd(a, m) = 1$.

Proof.

Theorem 1

An integer a has a multiplicative inverse modulo m iff $gcd(a, m) = 1$.

Proof.

($\Leftarrow$) Recall that there exist integers x and y such that

$$x \cdot a + y \cdot m = gcd(a, m) = 1.$$

Thus, $(x \cdot a + y \cdot m) \bmod m = x \cdot a \bmod m = 1 \bmod m$, i.e., $x \cdot a \equiv 1 \pmod{m}$. Therefore x is the inverse.

Theorem 1

An integer a has a multiplicative inverse modulo m iff $gcd(a, m) = 1$.

Proof.

($\Leftarrow$) Recall that there exist integers x and y such that

$$x \cdot a + y \cdot m = gcd(a, m) = 1.$$

Thus, $(x \cdot a + y \cdot m) \bmod m = x \cdot a \bmod m = 1 \bmod m$, i.e., $x \cdot a \equiv 1 \pmod{m}$. Therefore x is the inverse.

($\Rightarrow$) Let $r = gcd(a, m)$. Suppose that b is the multiplicative inverse of a modulo m, i.e., we have that

$$b \cdot a \equiv 1 \pmod{m},$$

Thus, $ba \bmod m = 1 \bmod m = 1$, i.e., there exists an integer q such that

$$ba = qm + 1,$$

or $ba - qm = 1$. However, r since $r|a$ and $r|m$, r also divides $bd - qm$ and 1. But it $r \not| 1$ because $r > 1$ and we have the contradiction. $\square$

Examples: division in modular arithmetic

Since the requirement for an existance of a^{-1} modulo m is that $gcd(a, m) = 1$, if we let m be a prime number, every a which is not a multiple of m has an inverse.
Can you solve this equation?

$$4x + 9 \equiv 0 \pmod{11}.$$

Examples: division in modular arithmetic

Since the requirement for an existance of a^{-1} modulo m is that $gcd(a, m) = 1$, if we let m be a prime number, every a which is not a multiple of m has an inverse.
Can you solve this equation?

$$4x + 9 \equiv 0 \pmod{11}.$$

We can even perform gaussian elimination (*which is very useful later*):

$$\begin{aligned} 2x + y &\equiv 3 \pmod 7 \\ x + 3y &\equiv 5 \pmod 7 \end{aligned}$$

Public-key cryptography

public key k_1

private key k_2

Alice

key

Eve

Bob

key

Bob: 121134

Private key cryptography

$E(M, k_1)$

$$D(E(M,k_1),k_2) \Rightarrow M$$

$k_1 = 121134$

$\boxed{k_2}$

Bob

M

RSA

Rivest Shamir Adelman

Public key (e, n) big number

Private key (d, n)

Message: m

- $\text{encrypt}(m) = (m^e) \bmod n$ %
- $\text{decrypt}(r) = (r^d) \bmod n$

Pick two prime numbers: p, q

$\boxed{n = pq}$

pick $\boxed{e}$ 65535

Calculate d: $\boxed{e^{-1} \pmod{(p-1)(q-1)}}$

RSA

$$(m^e) \bmod n$$

$$(a+b) \bmod n$$

$$=$$

$$((a \bmod n)+(b \bmod n)) \bmod n$$

$$(a \cdot b) \bmod n$$

$$=$$

$$((a \bmod n)(b \bmod n)) \bmod n$$

RSA: steps

- Private key: (d, n), Public key: (e, n)
- Encryption $E(m) = m^e \bmod n$, Decryption: $D(w) = w^d \bmod n$.
- Goal: Select e, d, n such that $D(E(m)) = m^{ed} \bmod n = m$.

RSA: steps

- Private key: (d, n), Public key: (e, n)
- Encryption $E(m) = m^e \bmod n$, Decryption: $D(w) = w^d \bmod n$.
- Goal: Select e, d, n such that $D(E(m)) = m^{ed} \bmod n = m$.

- Pick two primes p and q. Let $n = pq$.
- Pick e (usually a small number)
- Pick d such that $d = e^{-1} \pmod{(p-1)(q-1)}$, i.e., $ed \equiv 1 \pmod{(p-1)(q-1)}$, or

  $$ed = k \cdot (p-1)(q-1) + 1,$$

  for some integer k.
- What is $m^{ed} \bmod n$?

Secret sharing

Secret sharing scheme based on straight lines

Example: secret sharing

- Think of a secret number $m \in \{0, 1, \ldots, 10\}$.
- Pick a random number $a \in \{1, 2, \ldots, 10\}$.
- Your straight line function $f(x) = (ax + m) \bmod 11$.
- We will generate 3 points from f and give them to 3 of your friends, each with only 1 point. Pick 3 numbers x_1, x_2, x_3 from $\{1, 2, \ldots, 10\}$.
- Let's compute
$$(x_1, f(x_1)), \quad (x_2, f(x_2)), \quad (x_3, f(x_3)).$$
- Give them to 3 of your friends and challenge them to form a group of 2 people and figure out your number m.

What's next?

- We will prove Fermat's Little Theorem and show how to efficiently test if a number is prime.
- We will also use Fermat's Little Theorem to prove the correctness of RSA.
- Modular arithmetic is also key to our usage of polynomials to perform secret sharing and error correcting codes, because now we can do Gaussian elimination using only integers.