香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

Mathematics in OI (I)

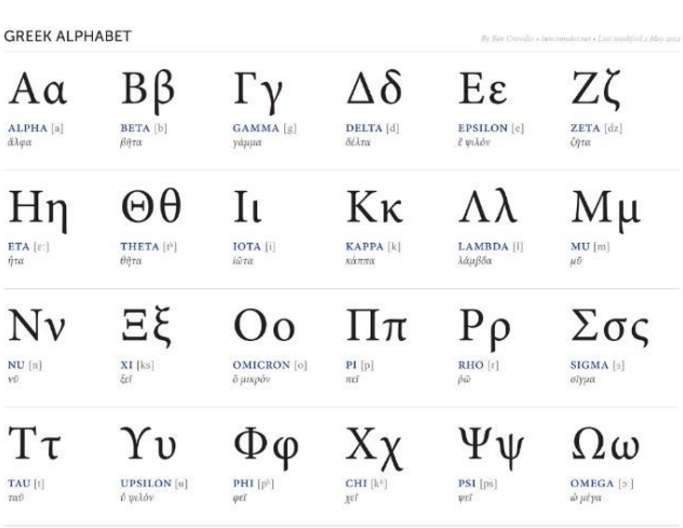
Daniel Hsieh {QwertyPi}

2026-04-10

Why Mathematics?

- We are dealing with numbers in our programs every day
  - Even characters are ASCII numbers!
- In OI, mathematics are everywhere
  - Many OI problems require mathematical knowledge to solve
  - The time complexity may be reduced by using some formulas
- Maths in OI (I): mainly cover number theory and modular arithmetic
- Maths in OI (II): mainly cover combinatorics

Table of Contents


GREEK ALPHABET
Αα
ALPHA [a]
ἄλφα
Ββ
BETA [b]
βῆτα
Γγ
GAMMA [g]
γάμμα
Δδ
DELTA [d]
δέλτα
Εε
EPSILON [e]
ἒ ψιλόν
Ζζ
ZETA [dz]
ζῆτα
Ηη
ETA [ɛ:]
ἦτα
Θθ
THETA [tʰ]
θῆτα
Ιι
IOTA [i]
ἰῶτα
Κκ
KAPPA [k]
κάππα
Λλ
LAMBDA [l]
λάμβδα
Μμ
MU [m]
μῦ
Νν
NU [n]
νῦ
Ξξ
XI [ks]
ξεῖ
Οο
OMICRON [o]
ὂ μικρόν
Ππ
PI [p]
πεῖ
Ρρ
RHO [r]
ῥῶ
Σσς
SIGMA [s]
σίγμα
Ττ
TAU [t]
ταῦ
Υυ
UPSILON [u]
ὖ ψιλόν
Φφ
PHI [pʰ]
φεῖ
Χχ
CHI [kʰ]
χεῖ
Ψψ
PSI [ps]
ψεῖ
Ωω
OMEGA [ɔ:]
ὦ μέγα

Some Notations and Abbreviations

- $\forall$: for all
- $\exists$: there exists
- $s.t.$: such that

- $\mathbb{N}$: set of natural numbers
- $\mathbb{Z}$: set of integers
- $\mathbb{Q}$: set of rational numbers
- $\mathbb{R}$: set of real numbers

- $\in$: is an element of
- $\cap$: set intersection
- $\cup$: set union

- $\wedge$: logical AND
- $\vee$: logical OR
- $\Rightarrow$: implies
- $\iff$: if and only if (iff)

Summation Sign Σ

Sigma: σ (lower case), Σ (upper case)
For any integers $l \leq r$,

$$\sum_{k=l}^{r} a_k = a_l + a_{l+1} + \cdots + a_r$$

Example:

$$\sum_{k=3}^{7} k^3 = 3^3 + 4^3 + 5^3 + 6^3 + 7^3 = 775$$

Summation Sign Σ

Another Example. Compute the following:

$$\sum_{k=20}^{30} (k^2 + 4k - 7)$$

Summation Sign Σ

Another Example. Compute the following:

$$\sum_{k=20}^{30} (k^2 + 4k - 7)$$

```
int sum = 0;
for (int k = 20; k <= 30; k++) {
    sum += k * k + 4 * k - 7;
}
return sum;
```

Product Sign Π

Pi: π (lower case), Π (upper case)
For any integers $l \leq r$,

$$\prod_{k=l}^{r} a_k = a_l \times a_{l+1} \times \cdots \times a_r$$

Example:

$$\prod_{k=3}^{7} (k^2 - 3) = (3^2 - 3)(4^2 - 3)(5^2 - 3)(6^2 - 3)(7^2 - 3) = 2604888$$

Product Sign Π

Probably most well-known example: Factorial, defined as

$$n! = \prod_{k=1}^{n} k = 1 \times 2 \times \cdots \times n$$

Note that $0! = 1$ by definition (in accordance with its combinatorical meaning)

In short, the summation sign (Σ) and product sign (Π) help you to express terms easier when lots of numbers are added / multiplied together.

Floor and Ceiling Function $\lfloor x \rfloor, \lceil x \rceil$

Floor function $\lfloor x \rfloor$ is defined to be the largest integer not exceeding x.
For example, $\lfloor 1 \rfloor = 1, \lfloor -2.718 \rfloor = -3, \lfloor \frac{10}{3} \rfloor = 3$.

Ceiling function $\lceil x \rceil$ is defined to be the smallest integer not less than x.
For example, $\lceil 1 \rceil = 1, \lceil -2.718 \rceil = -2, \lceil \frac{10}{3} \rceil = 4$.

Table of Contents

Divisibility Sign |

Vertical bar: |

$d \mid n$: "d divides n", "n is divisible by d" or "d is a factor of n".
$d \nmid n$ simply means the opposite, i.e., "n is not divisible by d".

Examples: $2 \mid 4, 3 \mid 6, 4 \mid 0, 6 \nmid 15$

But what does the statement "n is divisible by d" actually means?

Divisibility Sign |

Vertical bar: |

$d \mid n$: "d divides n", "n is divisible by d" or "d is a factor of n".
$d \nmid n$ simply means the opposite, i.e., "n is not divisible by d".

Examples: $2 \mid 4, 3 \mid 6, 4 \mid 0, 6 \nmid 15$

But what does the statement "n is divisible by d" actually means?

- In fact, $d \mid n$ is equivalent to that there exists an integer k such that $n = kd$.
- Formally,

$$d \mid n \iff \exists k \in \mathbb{Z} \text{ s.t. } n = kd$$

Divisibility Properties

Suppose a, b, c, x, y are all integers:

- $a \mid b$ and $b \mid c \Rightarrow a \mid c$
- $a \mid b$ and $a \mid c \Rightarrow a \mid (b \pm c)$
- $a \mid b$ and $b \mid a \Rightarrow a = \pm b$
- $a \mid b \Rightarrow a \mid bx$
- $a \mid b \Rightarrow ax \mid bx$
- $a \mid b$ and $a \mid c \Rightarrow a \mid (bx + cy)$

All these can be deduced from the definitions on them own.

Table of Contents

Greatest Common Divisor (GCD) - Definition

Also known as Highest Common Factor (HCF)

Consider two integers a and b.

- If d divides both a and b, then d is a common divisor of a and b.

Greatest Common Divisor (GCD) - Definition

Greatest common divisor (GCD) of a and b is simply the largest among the common divisors of a and b.

- We use $\gcd(a, b)$ to represent the greatest common divisor of a and b.
- You can also use (a, b) when no confusion would arise (with coordinates).

We say two integers a, b are co-prime (or relatively prime) if $\gcd(a, b) = 1$.

Greatest Common Divisor (GCD) - Computation

How can we calculate GCD of two integers?

1. Brute force! Too slow :(

Greatest Common Divisor (GCD) - Computation

How can we calculate GCD of two integers?

1. Brute force! Too slow :(
2. Use C++17! But how does it work?

Greatest Common Divisor (GCD) - Computation

How can we calculate GCD of two integers?

1. Brute force! Too slow :(
2. Use C++17! But how does it work?
3. Euclidean Algorithm $\hookleftarrow$ Our target

What do we know about GCD? What observations can we make?

Euclidean Algorithm - Concept

Observation 1

$\gcd(a, b) = \gcd(b, a)$, i.e. the order does not matter.

Euclidean Algorithm - Concept

Observation 1

$\gcd(a, b) = \gcd(b, a)$, i.e. the order does not matter.

Observation 2

For any positive integer a, $\gcd(a, 0) = a$.

Euclidean Algorithm - Concept

Observation 3

For any integers a, b and k, $\gcd(a, b) = \gcd(a, b + ka)$ holds.

Euclidean Algorithm - Concept

Observation 3

For any integers a, b and k, $\gcd(a, b) = \gcd(a, b + ka)$ holds.

Observation 4

For any integers $a, b (b \neq 0)$, $\gcd(a, b) = \gcd(b, a \bmod b)$ holds.

Euclidean Algorithm - Procedure

We have introduced all the building blocks of Euclidean Algorithm!

Euclidean Algorithm

Given two positive integers a and b. To find their greatest common divisor, we can

- Apply the fact that $\gcd(a, b) = \gcd(b, a \bmod b)$ repeatedly until the second parameter becomes 0.
- The answer is then simply the first parameter.

Euclidean Algorithm - Procedure

The code for Euclidean Algorithm is incredibly simple:

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
```

Euclidean Algorithm - Procedure

The code for Euclidean Algorithm is incredibly simple:

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
```

Now, two questions for you:

1. Why is it guaranteed to terminate?
2. When does this algorithm perform the worst? (It is $O(\log \min(a, b))$)

Euclidean Algorithm - Example

a	b	r	q
2026	411		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$2026 = 411 \times 4 + 382$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$2026 = 411 \times 4 + 382$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$411 = 382 \times 1 + 29$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$411 = 382 \times 1 + 29$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$382 = 29 \times 13 + 5$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$382 = 29 \times 13 + 5$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$29 = 5 \times 5 + 4$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$29 = 5 \times 5 + 4$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4	1	1

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$5 = 4 \times 1 + 1$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4	1	1
4	1		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$5 = 4 \times 1 + 1$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4	1	1
4	1	0	4

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$4 = 1 \times 4 + 0$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4	1	1
4	1	0	4
1	0		

Today
2026-04-11

Goal
Find $\gcd(2026, 411)$

$$4 = 1 \times 4 + 0$$

Euclidean Algorithm - Example

a	b	r	q
2026	411	382	4
411	382	29	1
382	29	5	13
29	5	4	5
5	4	1	1
4	1	0	4
1	0		

Today
2026-04-11

$\gcd(2026, 411) = 1$

So that's all for Euclidean Algorithm - let's move back to GCD!

Greatest Common Divisor (GCD) - Properties

Property 1

If m is a common divisor of a and b, then $m \mid \gcd(a, b)$.

Greatest Common Divisor (GCD) - Properties

Property 1

If m is a common divisor of a and b, then $m \mid \gcd(a, b)$.

Property 2

$\gcd(ka, kb) = k \times \gcd(a, b)$

Greatest Common Divisor (GCD) - Properties

Property 3

Let d denote the greatest common divisor of a and b. Then

$$\gcd\left(\frac{a}{d}, \frac{b}{d}\right) = 1$$

Greatest Common Divisor (GCD) - Properties

Property 3

Let d denote the greatest common divisor of a and b. Then

$$\gcd\left(\frac{a}{d}, \frac{b}{d}\right) = 1$$

Property 4

Let d denote the greatest common divisor of a and b. If k divides d, then

$$\gcd\left(\frac{a}{k}, \frac{b}{k}\right) = \frac{\gcd(a, b)}{k}$$

Greatest Common Divisor (GCD) - >2 Numbers

What is $\gcd(a_1, a_2, \ldots, a_N)$?

Greatest Common Divisor (GCD) - >2 Numbers

What is $\gcd(a_1, a_2, \ldots, a_N)$?

As the order doesn't matter, we can simply calculate them one by one:

$$\gcd(a_1, a_2, \ldots, a_N) = \gcd(\gcd(\ldots \gcd(\gcd(a_1, a_2), a_3) \cdots a_{N-1}), a_N)$$

Greatest Common Divisor (GCD) - Application

1. Simplifying a fraction $\frac{a}{b}$:

$$\frac{a}{b} = \frac{a \div \gcd(a, b)}{b \div \gcd(a, b)}$$

Greatest Common Divisor (GCD) - Application

1. Simplifying a fraction $\frac{a}{b}$:

$$\frac{a}{b} = \frac{a \div \gcd(a,b)}{b \div \gcd(a,b)}$$

2. Solving linear Diophantine equation for integers x and y:
   - $ax + by = c$ where $\gcd(a,b) \mid c$
   - can be solved by Extended Euclidean Algorithm

Greatest Common Divisor (GCD) - Application

1. Simplifying a fraction $\frac{a}{b}$:

$$\frac{a}{b} = \frac{a \div \gcd(a,b)}{b \div \gcd(a,b)}$$

2. Solving linear Diophantine equation for integers x and y:
   - $ax + by = c$ where $\gcd(a,b) \mid c$
   - can be solved by Extended Euclidean Algorithm

3. Calculating the least common multiple (LCM):
   - $\text{lcm}(a,b)$ is defined as the *least common* multiple of integers a and b.
   - Can be calculated as $\text{lcm}(a,b) = \frac{ab}{\gcd(a,b)}$
   - To be briefly introduced in the next few slides

Least Common Multiple (LCM) - Definition

Consider two integers a and b.

- If both a and b divides m, then m is a common multiple of a and b.
- Symbolic form: $a \mid m \wedge b \mid m$ or simply $a, b \mid m$
- Least common multiple (LCM) of a and b is simply the smallest among the common multiples of a and b.
- We use $\text{lcm}(a, b)$ to represent the least common multiple of a and b.
- You can also use $[a, b]$ when no confusion would arise.

Least Common Multiple (LCM) - Properties

Property 1

if n is a common multiple of a and b, then $\text{lcm}(a, b) \mid n$.

Least Common Multiple (LCM) - Properties

Property 1

if n is a common multiple of a and b, then $\mathsf{lcm}(a, b) \mid n$.

Property 2

$\mathsf{lcm}(ka, kb) = k \times \mathsf{lcm}(a, b)$

Least Common Multiple (LCM) - Properties

Property 1

if n is a common multiple of a and b, then $\mathrm{lcm}(a, b) \mid n$.

Property 2

$\mathrm{lcm}(ka, kb) = k \times \mathrm{lcm}(a, b)$

Property 3

For any positive integers a, b, we have $ab = \gcd(a, b) \times \mathrm{lcm}(a, b)$.

Least Common Multiple (LCM) - >2 Numbers

What is lcm$(a_1, a_2, \ldots, a_N)$?

Least Common Multiple (LCM) - >2 Numbers

What is $\text{lcm}(a_1, a_2, \ldots, a_N)$?

As the order doesn't matter, we can simply calculate them one by one:

$$\text{lcm}(a_1, a_2, \ldots, a_N) = \text{lcm}(\text{lcm}(\ldots \text{lcm}(\text{lcm}(a_1, a_2), a_3) \cdots a_{N-1}), a_N)$$

Table of Contents

Modular Arithmetic - Introduction

Recall what you have learnt in primary about division:
$17 \div 5 = 3 \ldots 2$ (Dividend $\div$ Divisor $=$ Quotient $\ldots$ Reminder)
So we say the remainder of $17 \div 5$ is 2

Modular Arithmetic - Introduction

Recall what you have learnt in primary about division:
$17 \div 5 = 3 \ldots 2$ (Dividend $\div$ Divisor $=$ Quotient $\ldots$ Reminder)
So we say the remainder of $17 \div 5$ is 2

Now, we may say $17 \equiv 2 \pmod 5$
"17 and 2 are congruent modulo 5"

Modular Arithmetic - Notation

Generally, if a and b are congruent modulo m where a, b are integers, m is a positive integer, we write $a \equiv b \pmod{m}$

Translate into programming language:

- Pascal: `a mod m = b mod m`
- C++: `a % m == b % m`

The remainder of the division of both a and b by m are the same

Modular Arithmetic - Caution

Take care of the sign of the dividend when writing programs.
If the dividend is negative, the remainder becomes non-positive!

For example, $-5 \equiv 2 \pmod 7$, but `-5 % 7` gives result `-5`

Modular Arithmetic - Caution

Take care of the sign of the dividend when writing programs.
If the dividend is negative, the remainder becomes non-positive!

For example, $-5 \equiv 2 \pmod 7$, but `-5 % 7` gives result `-5`

Hence, you may need to write this:

```
((a % b) + b) % b
```

Modular Arithmetic - Meaning

Instead of saying a and b have the same remainder when divided by m, we can say that:

Modular Arithmetic - Meaning

Instead of saying a and b have the same remainder when divided by m, we can say that:

1. $a - b$ is divisible by m.
   Mathematical Notation: $m \mid (a - b)$

Modular Arithmetic - Meaning

Instead of saying a and b have the same remainder when divided by m, we can say that:

1. $a - b$ is divisible by m.
   Mathematical Notation: $m \mid (a - b)$
2. There exists some integer k such that $a = km + b$.
   Mathematical Notation: $\exists k \in \mathbb{Z}$ s.t. $a = km + b$

Modular Arithmetic - Addition / Subtraction / Multiplication

If $a \equiv b \pmod{m}, c \equiv d \pmod{m}$, then for $x \in \mathbb{Z}$,

Modular Arithmetic - Addition / Subtraction / Multiplication

If $a \equiv b \pmod{m}, c \equiv d \pmod{m}$, then for $x \in \mathbb{Z}$,

- $a \pm x \equiv b \pm x \pmod{m}$
- $a \pm c \equiv b \pm d \pmod{m}$

Modular Arithmetic - Addition / Subtraction / Multiplication

If $a \equiv b \pmod{m}, c \equiv d \pmod{m}$, then for $x \in \mathbb{Z}$,

- $a \pm x \equiv b \pm x \pmod{m}$
- $a \pm c \equiv b \pm d \pmod{m}$
- $ax \equiv bx \pmod{m}$
- $ax \equiv bx \pmod{mx}$ if $x > 0$
- $ac \equiv bd \pmod{m}$

Modular Arithmetic - Addition / Subtraction / Multiplication

If $a \equiv b \pmod{m}, c \equiv d \pmod{m}$, then for $x \in \mathbb{Z}$,

- $a \pm x \equiv b \pm x \pmod{m}$
- $a \pm c \equiv b \pm d \pmod{m}$
- $ax \equiv bx \pmod{m}$
- $ax \equiv bx \pmod{mx}$ if $x > 0$
- $ac \equiv bd \pmod{m}$

Proof. Leave as exercise for readers.
(Hint: You can show $m \mid (LHS - RHS)$) □

Modular Arithmetic - Addition / Subtraction / Multiplication

If $a \equiv b \pmod{m}, c \equiv d \pmod{m}$, then for $x \in \mathbb{Z}$,

- $a \pm x \equiv b \pm x \pmod{m}$
- $a \pm c \equiv b \pm d \pmod{m}$
- $ax \equiv bx \pmod{m}$
- $ax \equiv bx \pmod{mx}$ if $x > 0$
- $ac \equiv bd \pmod{m}$

Proof. Leave as exercise for readers.
(Hint: You can show $m \mid (LHS - RHS)$) □

This basically means when we are doing $+, -, \times$ under modulo, we can replace any number with anything else, as long as they are congruent modulo m.

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Sadly, that is not the case. Let's illustrate the issues with some examples:

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Sadly, that is not the case. Let's illustrate the issues with some examples:

1. What is the value of $4 \div 2 \pmod{6}$?

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Sadly, that is not the case. Let's illustrate the issues with some examples:

1. What is the value of $4 \div 2 \pmod 6$?
2. What is the value of $3 \div 5 \pmod 7$?

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Sadly, that is not the case. Let's illustrate the issues with some examples:

1. What is the value of $4 \div 2 \pmod 6$?
2. What is the value of $3 \div 5 \pmod 7$?

For (1), you probably will say $2 \pmod 6$ - but $5 \pmod 6$ is also a possible value.

For (2), even it is a 'fraction', it has the value of $2 \pmod 7$.

Modular Division - Introduction

How about division? Can we carry out division as 'normal' arithmetic also?

Sadly, that is not the case. Let's illustrate the issues with some examples:

1. What is the value of $4 \div 2 \pmod 6$?
2. What is the value of $3 \div 5 \pmod 7$?

For (1), you probably will say $2 \pmod 6$ - but $5 \pmod 6$ is also a possible value.

For (2), even it is a 'fraction', it has the value of $2 \pmod 7$.

Problem: What actually *is* a modular division anyways?

Modular Division - Definition

Definition of Modular Division

Let a, b, m be integers which $b \neq 0$ and m is positive.
Then, $a \div b$ modulo m is defined to be the integral solutions x for

$$bx \equiv a \pmod{m}$$

Modular Division - Definition

Definition of Modular Division

Let a, b, m be integers which $b \neq 0$ and m is positive.
Then, $a \div b$ modulo m is defined to be the integral solutions x for

$$bx \equiv a \pmod{m}$$

For example, $3 \div 5 \equiv 2 \pmod{7}$ as $2 \times 5 = 10 \equiv 3 \pmod{7}$.

Modular Division - Definition

Definition of Modular Division

Let a, b, m be integers which $b \neq 0$ and m is positive.
Then, $a \div b$ modulo m is defined to be the integral solutions x for

$$bx \equiv a \pmod{m}$$

For example, $3 \div 5 \equiv 2 \pmod{7}$ as $2 \times 5 = 10 \equiv 3 \pmod{7}$.
Meanwhile, $4 \div 2 \equiv 2, 5 \pmod{6}$ or $2 \pmod{3}$ as $2 \times 2 \equiv 5 \times 2 \equiv 4 \pmod{6}$.

Modular Division - Definition

Definition of Modular Division

Let a, b, m be integers which $b \neq 0$ and m is positive.
Then, $a \div b$ modulo m is defined to be the integral solutions x for

$$bx \equiv a \pmod{m}$$

For example, $3 \div 5 \equiv 2 \pmod{7}$ as $2 \times 5 = 10 \equiv 3 \pmod{7}$.
Meanwhile, $4 \div 2 \equiv 2, 5 \pmod{6}$ or $2 \pmod{3}$ as $2 \times 2 \equiv 5 \times 2 \equiv 4 \pmod{6}$.

More questions:

1. When does it have solutions?
2. How can we find solutions *quickly* when it exists?

Modular Inverse - Definition

Modular Inverse is another concept related to division in modular arithmetic:

Definition of Modular Inverse

Let a, m be integers which $a \neq 0$ and m is positive.
Then, a^{-1} modulo m is defined to be the integral solutions x for

$$ax \equiv 1 \pmod{m}$$

Modular Inverse - Definition

Modular Inverse is another concept related to division in modular arithmetic:

Definition of Modular Inverse

Let a, m be integers which $a \neq 0$ and m is positive.
Then, a^{-1} modulo m is defined to be the integral solutions x for

$$ax \equiv 1 \pmod{m}$$

Indeed, we can do modular division with modular inverse (when it exists):

$$ax \equiv c \pmod{m} \iff a^{-1}ax \equiv x \equiv a^{-1}c \pmod{m}$$

Modular Inverse - Existence

For some integer a and positive integer m, modular inverse might not exist.

- If $a = 6, m = 4$, for any integer b, ab is always even.
- That is, $ab \equiv 0$ or $2 \pmod 4 \Rightarrow ab \not\equiv 1 \pmod 4$

Modular Inverse - Existence

For some integer a and positive integer m, modular inverse might not exist.

- If $a = 6, m = 4$, for any integer b, ab is always even.
- That is, $ab \equiv 0$ or $2 \pmod 4 \Rightarrow ab \not\equiv 1 \pmod 4$

Modular inverse only exists if $\gcd(a, m) = 1$, that is, a, m are relatively prime.

- Notably, when m is a *prime* while a is not a multiple of m, then modular inverse of a exists.

Modular Inverse - Existence

For some integer a and positive integer m, modular inverse might not exist.

- If $a = 6, m = 4$, for any integer b, ab is always even.
- That is, $ab \equiv 0$ or $2 \pmod 4 \Rightarrow ab \not\equiv 1 \pmod 4$

Modular inverse only exists if $\gcd(a, m) = 1$, that is, a, m are relatively prime.

- Notably, when m is a *prime* while a is not a multiple of m, then modular inverse of a exists.

Note that a *prime* as a positive integer that has exactly two positive divisors, namely 1 and itself.

Modular Inverse - Reality

In fact, most of the modulo (except perhaps in the problem set) you will ever have to deal with in OI problems are primes.

Most common examples:

- $10^9 + 7$
- $998244353 = 7 \times 17 \times 2^{23} + 1$

Modular Inverse - Reality

In fact, most of the modulo (except perhaps in the problem set) you will ever have to deal with in OI problems are primes.

Most common examples:

- $10^9 + 7$
- $998244353 = 7 \times 17 \times 2^{23} + 1$

Majority of time, modulo are there simply to make the answer small enough to fit in 32-bit integer. It is less relevant to the problem itself.

Modular Inverse - Reality

In fact, most of the modulo (except perhaps in the problem set) you will ever have to deal with in OI problems are primes.

Most common examples:

- $10^9 + 7$
- $998244353 = 7 \times 17 \times 2^{23} + 1$

Majority of time, modulo are there simply to make the answer small enough to fit in 32-bit integer. It is less relevant to the problem itself.

Therefore, most likely when you have to calculate modular inverse, the modulo is a prime.

Modular Inverse - Computation

Two main ways can be used to calculate modular inverse for prime modulo p.

Modular Inverse - Computation

Two main ways can be used to calculate modular inverse for prime modulo p.

1. Extended Euclidean Algorithm - solve $ax + kp = 1$.
   - Technically also works for non-prime modulo.

Modular Inverse - Computation

Two main ways can be used to calculate modular inverse for prime modulo p.

1. Extended Euclidean Algorithm - solve $ax + kp = 1$.
   - Technically also works for non-prime modulo.

2. Fermat's Little Theorem - (when $p \nmid a$) $a^{-1} \equiv a^{p-2} \pmod{p}$. $\hookleftarrow$ Our target
   - Can also be extended for non-prime modulo using the Euler's totient function φ.

Modular Inverse - Fermat's Little Theorem

Fermat's Little Theorem

Given integer a and prime p which $p \nmid a$. Then

$$a^{p-1} \equiv 1 \pmod{p}$$

Modular Inverse - Fermat's Little Theorem

Fermat's Little Theorem

Given integer a and prime p which $p \nmid a$. Then

$$a^{p-1} \equiv 1 \pmod{p}$$

Proof. Notice that

$$a \times 2a \times \cdots \times (p-1)a \equiv 1 \times 2 \times \cdots \times (p-1) \pmod{p}$$

as $\{a, 2a, \ldots, (p-1)a\}$ is simply a permutation of $\{1, 2, \ldots, p-1\}$ modulo p (Why?). By canceling $(p-1)! = 1 \times 2 \times \cdots \times (p-1)$ on both sides, we obtain

$$a^{p-1} \equiv 1 \pmod{p}$$

Modular Inverse - Fermat's Little Theorem

Therefore, for integer a such that $p \nmid a$, we have

$$a \times a^{p-2} \equiv a^{p-1} \equiv 1 \pmod{p}$$

Modular Inverse - Fermat's Little Theorem

Therefore, for integer a such that $p \nmid a$, we have

$$a \times a^{p-2} \equiv a^{p-1} \equiv 1 \pmod{p}$$

which means a^{p-2} is the modular inverse of a!

Modular Inverse - Fermat's Little Theorem

Therefore, for integer a such that $p \nmid a$, we have

$$a \times a^{p-2} \equiv a^{p-1} \equiv 1 \pmod{p}$$

which means a^{p-2} is the modular inverse of a!

Here comes the problem: How can we compute exponential quick enough?

Fast Exponential - Problem

Given a, b, m, find $a^b \pmod{m}$.

Fast Exponential - Problem

Given a, b, m, find $a^b \pmod{m}$.

Code #1:

```
int product = 1;
for (int i = 1; i <= b; i++)
    product = (product * a) % m;
return product;
```

Fast Exponential - Problem

Given a, b, m, find $a^b \pmod{m}$.

Code #1:

```
int product = 1;
for (int i = 1; i <= b; i++)
    product = (product * a) % m;
return product;
```

Two problems:

1. Time Complexity: $O(b)$, not fast enough for large b

Fast Exponential - Problem

Given a, b, m, find $a^b \pmod{m}$.

Code #1:

```
int product = 1;
for (int i = 1; i <= b; i++)
    product = (product * a) % m;
return product;
```

Two problems:

1. Time Complexity: $O(b)$, not fast enough for large b
2. Also, use `long long`!

Fast Exponential - Big Mod

You will first need to know that the law of indices:

$$a^{n+m} = a^n \times a^m$$

Then, there are two methods to proceed:

1. Iterative method
2. Recursive method

Fast Exponential - Big Mod (Iterative)

We can express b as sum of power of 2 as

$$b = 2^{s_1} + 2^{s_2} + \cdots + 2^{s_k}$$

which $s_1 < s_2 < \cdots < s_k$.
Then,

$$a^b = a^{\sum_{i=1}^{k} 2^{s_i}} = \prod_{i=1}^{k} a^{2^{s_i}}$$

Time Complexity: $O(\log n)$.

Fast Exponential - Big Mod (Iterative)

Example: $7^{77} \mod 23 \equiv 22, 77_{10} = 1001101_2$

$$
\begin{aligned}
7^1 &\equiv 7 \pmod{23} \\
7^2 &\equiv 3 \pmod{23} \\
7^4 &\equiv 3^2 \equiv 9 \pmod{23} \\
7^8 &\equiv 9^2 \equiv 12 \pmod{23} \\
7^{16} &\equiv 12^2 \equiv 6 \pmod{23} \\
7^{32} &\equiv 6^2 \equiv 13 \pmod{23} \\
7^{64} &\equiv 13^2 \equiv 8 \pmod{23}
\end{aligned}
$$

$$
\begin{aligned}
7^{77} &\equiv 7^{64+8+4+1} \\
&\equiv 8 \times 12 \times 9 \times 7 \\
&\equiv 22 \pmod{23}
\end{aligned}
$$

Fast Exponential - Big Mod (Recursive)

We can also recursively calculate the answer.

- If $b = 0$, then $a^b \equiv 1 \pmod{m}$.
- If $b = 2k$, then $a^b \equiv a^k \times a^k \pmod{m}$.
- If $b = 2k + 1$, then $a^b \equiv a^k \times a^k \times a \pmod{m}$.

Fast Exponential - Big Mod (Recursive)

We can also recursively calculate the answer.

- If $b = 0$, then $a^b \equiv 1 \pmod{m}$.
- If $b = 2k$, then $a^b \equiv a^k \times a^k \pmod{m}$.
- If $b = 2k + 1$, then $a^b \equiv a^k \times a^k \times a \pmod{m}$.

Therefore, we can reduce the problem of finding $a^b \pmod{m}$ to $a^{\lfloor \frac{b}{2} \rfloor} \pmod{m}$.
Time Complexity: $O(\log n)$.

Actually both methods is not limited to modular exponential - and can be used for other stuff as long as the operation are associative.

We obtain yet another method to calculate $a^{-1} \equiv a^{p-2} \pmod{p}$, p prime.

Table of Contents

Prime Number - Definition

Prime numbers are positive integers which has exactly 2 positive divisors.

- The first few prime numbers are $2, 3, 5, 7, 11, 13, \ldots$

Composite numbers are positive integers with more than 2 positive divisors.

- The first few composite numbers are $4, 6, 8, 9, 10, 12, \ldots$.

Note that 1 is neither a prime nor a composite number.

Primality Test - Brute force

How can we check whether a positive integer is prime or not?

Primality Test - Brute force

How can we check whether a positive integer is prime or not?

Observation 1

For each prime p, it has exactly 2 positive divisors, namely 1 and p.

Primality Test - Brute force

How can we check whether a positive integer is prime or not?

Observation 1

For each prime p, it has exactly 2 positive divisors, namely 1 and p.

Code 1:

```
for (int i = 2; i < n; i++)
    if (n % i == 0)
        return false;
return true
```

Time Complexity: $O(n)$:(

Primality Test - Brute Force Till $\sqrt{n}$

Observation 2

If n is composite, there exists a divisor d of n such that $1 < d \leq \sqrt{n}$.

Code 2:

```
for (int i = 2; i * i <= n; i++) // must be <=, not <
    if (n % i == 0)
        return false;
return true
```

Time Complexity: $O(\sqrt{n})$:)

Prime Sieve - Brute Force Till $\sqrt{n}$

Now - how about finding ALL the prime numbers between 1 and 10^6?

Prime Sieve - Brute Force Till $\sqrt{n}$

Now - how about finding ALL the prime numbers between 1 and 10^6?

Alternatively, how can we *sieve away* all the composite numbers?

Prime Sieve - Brute Force Till $\sqrt{n}$

Now - how about finding ALL the prime numbers between 1 and 10^6?

Alternatively, how can we *sieve away* all the composite numbers?

Code 2*:

```
for (int j = 1; j <= 1000000; j++)
    if (is_prime(j)) output j; // Using Code #2
```

Time Complexity: $O(n\sqrt{n})$:(

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

- Consider its smallest divisor $p > 1$. We want to show that p must be prime.

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

- Consider its smallest divisor $p > 1$. We want to show that p must be prime.
- Suppose otherwise, i.e. p is composite.

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

- Consider its smallest divisor $p > 1$. We want to show that p must be prime.
- Suppose otherwise, i.e. p is composite.
- Then we can find a divisor of p that is neither 1 and p, say a.

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

- Consider its smallest divisor $p > 1$. We want to show that p must be prime.
- Suppose otherwise, i.e. p is composite.
- Then we can find a divisor of p that is neither 1 and p, say a.
- But then $a \mid n$ also, contradicting the fact that p is the smallest divisor (greater than 1) of n.

Prime Sieve - Sieve of Eratosthenes

Observation 3

For any composite number, it is divisible by some prime number.
For any prime number, the only prime divisor is itself.

Proof. For composite numbers n,

- Consider its smallest divisor $p > 1$. We want to show that p must be prime.
- Suppose otherwise, i.e. p is composite.
- Then we can find a divisor of p that is neither 1 and p, say a.
- But then $a \mid n$ also, contradicting the fact that p is the smallest divisor (greater than 1) of n.

The prime number part is straight from definition.

Prime Sieve - Sieve of Eratosthenes

Hence, we may store a list of prime numbers we have found.

- For each integer, check if it is divided by any of the prime numbers found.
- If not, then it is a prime, which we can add it to the prime list.
- We may use store a Boolean array which indicates whether each integer is known to be composite.

Prime Sieve - Sieve of Eratosthenes

Current Step: List out all the numbers													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate 1													
	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 2 except 2													
	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 2 except 2

	2	3		5		7		9		11		13	
15		17		19		21		23		25		27	
29		31		33		35		37		39		41	
43		45		47		49		51		53		55	
57		59		61		63		65		67		69	
71		73		75		77		79		81		83	
85		87		89		91		93		95		97	
99		101		103		105		107		109		111	
113		115		117		119		121		123		125	
127		129		131		133		135		137		139	
141		143		145		147		149		151		153	

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 3 except 3											
	2	3		5		7		9		11		13	
15		17		19		21		23		25		27	
29		31		33		35		37		39		41	
43		45		47		49		51		53		55	
57		59		61		63		65		67		69	
71		73		75		77		79		81		83	
85		87		89		91		93		95		97	
99		101		103		105		107		109		111	
113		115		117		119		121		123		125	
127		129		131		133		135		137		139	
141		143		145		147		149		151		153	

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 3 except 3												
	2	3		5		7				11		13
		17		19				23		25		
29		31				35		37				41
43				47		49				53		55
		59		61				65		67		
71		73				77		79				83
85				89		91				95		97
		101		103				107		109		
113		115				119		121				125
127				131		133				137		139
		143		145				149		151		

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 5 except 5

	2	3		5		7				11		13
		17		19				23		25		
29		31				35		37				41
43				47		49				53		55
		59		61				65		67		
71		73				77		79				83
85				89		91				95		97
		101		103				107		109		
113		115				119		121				125
127				131		133				137		139
		143		145				149		151		

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 5 except 5

	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47		49				53			
		59		61						67			
71		73				77		79				83	
				89		91						97	
		101		103				107		109			
113						119		121					
127				131		133				137		139	
		143						149		151			

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 7 except 7

	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47		49				53			
		59		61						67			
71		73				77		79				83	
				89		91						97	
		101		103				107		109			
113						119		121					
127				131		133				137		139	
		143						149		151			

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 7 except 7													
	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47						53			
		59		61						67			
71		73						79				83	
				89								97	
		101		103				107		109			
113								121					
127				131						137		139	
		143						149		151			

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 11 except 11

	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47						53			
		59		61						67			
71		73						79				83	
				89								97	
		101		103				107		109			
113								121					
127				131						137		139	
		143						149		151			

Prime Sieve - Sieve of Eratosthenes

Current Step: Eliminate multiples of 11 except 11

	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47						53			
		59		61						67			
71		73						79				83	
				89								97	
		101		103				107		109			
113													
127				131						137		139	
								149		151			

Prime Sieve - Sieve of Eratosthenes

Current Step: Done since $13 \times 13 > 154$

	2	3		5		7				11		13	
		17		19				23					
29		31						37				41	
43				47						53			
		59		61						67			
71		73						79				83	
				89								97	
		101		103				107		109			
113													
127				131						137		139	
								149		151			

Prime Sieve - Sieve of Eratosthenes

```
// initialise boolean array comp[] with false;
for (int i = 2; i * i <= n; i++) {
    if (comp[i] == false) {
        for (int j = i * i; j <= n; j += i)
            comp[j] = true;
    }
}
```

Time Complexity: $O(n \log \log n)$

Table of Contents

Prime Factorisation - Definition

Fundamental Theorem of Arithmetic

For any positive integer n, we can represent n uniquely as

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

which $p_1 < p_2 < \cdots < p_k$ are primes while $\alpha_1, \alpha_2, \ldots, \alpha_k$ are positive integers.

Notice that above is actually talking about two points at the same time:

1. Existence. There exist such a way to write n as products of primes.

Prime Factorisation - Definition

Fundamental Theorem of Arithmetic

For any positive integer n, we can represent n uniquely as

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

which $p_1 < p_2 < \cdots < p_k$ are primes while $\alpha_1, \alpha_2, \ldots, \alpha_k$ are positive integers.

Notice that above is actually talking about two points at the same time:

1. Existence. There exist such a way to write n as products of primes.
2. Uniqueness. There does not exists two different ways to represent n.

Prime Factorisation - Using Sieve

We can modify the Sieve of Eratosthenes to maintain smallest prime factor:

- Replace boolean array with integer array.
- When we mark an integer as composite, store the current prime factor.

Prime Factorisation - Using Sieve

We can modify the Sieve of Eratosthenes to maintain smallest prime factor:

- Replace boolean array with integer array.
- When we mark an integer as composite, store the current prime factor.

Then, for each integer, we can get the smallest prime divisor instantly, and we can factorise the number recursively.

Factorial Factors - More Observations

Example: $n = 20, k = 2$																			
Divisibility	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
4				✓				✓				✓				✓				✓
8								✓								✓				
16																✓				
Highest Power	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4	0	1	0	2
Sum of Highest Powers = 18																				

Factorial Factors - More Observations

Example: $n = 20, k = 2$																				
Divisibility	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
4				✓				✓				✓				✓				✓
8								✓								✓				
16																✓				
Highest Power	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4	0	1	0	2
Sum of Highest Powers = 18																				

Is there another way to come up the result 18?
(Or equivalently, any other way to count the number of ticks?)

Factorial Factors - More Observations

Example: $n = 20, k = 2$																			
Divisibility	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
4				✓				✓				✓				✓				✓
8								✓								✓				
16																✓				
Highest Power	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4	0	1	0	2
Sum of Highest Powers = 18																				

Is there another way to come up the result 18?
(Or equivalently, any other way to count the number of ticks?)

Instead of counting the number of ticks per column,
try counting it per row, which is also meaningful!

Factorial Factors - More Observation

Row r repersents: the number of integers between 1 and n in which they are divisible by p^r.

How many? It's easy! :)

Factorial Factors - More Observation

Row r repersents: the number of integers between 1 and n in which they are divisible by p^r.

How many? It's easy! :) Answer: $\left\lfloor \frac{n}{p^r} \right\rfloor$

Factorial Factors - Implementation

Code #2:

```
int k = 0;
while (n) {
    n /= p;
    k += n;
}
return k;
```

Time Complexity: $O(\log n)$

Table of Contents

Addition and Multiplication Principle

Addition Principle

If we want to choose 1 element out of k given sets with sizes $n_1, n_2, \cdots, n_k$, and no two sets share a common element, then there are in total $n_1 + n_2 + \cdots + n_k$ ways to choose an element.

Addition and Multiplication Principle

Addition Principle

If we want to choose 1 element out of k given sets with sizes $n_1, n_2, \cdots, n_k$, and no two sets share a common element, then there are in total $n_1 + n_2 + \cdots + n_k$ ways to choose an element.

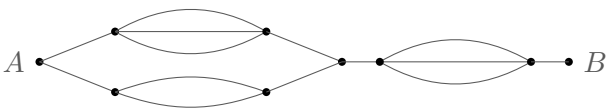
Multiplication Principle

If we want to choose 1 element out of **each** of the k given sets with sizes $n_1, n_2, \cdots, n_k$, then there are in total $n_1 \times n_2 \times \cdots \times n_k$ ways to choose the elements.

Addition and Multiplication Principle

Poll: How many paths are there from A to B?

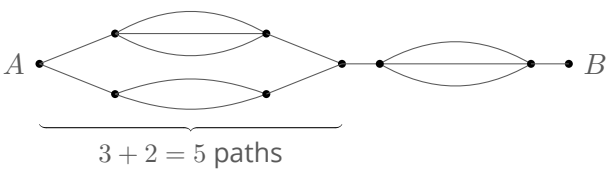
A. $(3 + 2) + 3$ paths
B. $(3 + 2) \times 3$ paths
C. $(3 \times 2) + 3$ paths
D. $(3 \times 2) \times 3$ paths



Addition and Multiplication Principle

Poll: How many paths are there from A to B?

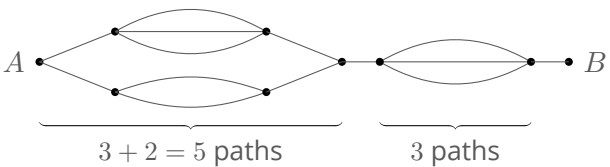
A. $(3 + 2) + 3$ paths
B. $(3 + 2) \times 3$ paths
C. $(3 \times 2) + 3$ paths
D. $(3 \times 2) \times 3$ paths



Addition and Multiplication Principle

Poll: How many paths are there from A to B?

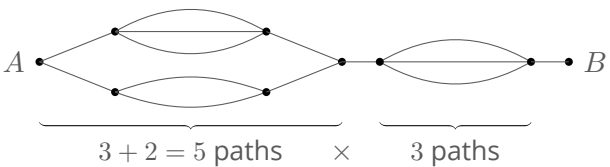
A. $(3 + 2) + 3$ paths
B. $(3 + 2) \times 3$ paths
C. $(3 \times 2) + 3$ paths
D. $(3 \times 2) \times 3$ paths



Addition and Multiplication Principle

Poll: How many paths are there from A to B?

A. $(3 + 2) + 3$ paths
B. $(3 + 2) \times 3$ paths
C. $(3 \times 2) + 3$ paths
D. $(3 \times 2) \times 3$ paths



Factorial

$n!$ (n factorial)

Factorial

$n!$ (n factorial)

Recursive definition: $n! = \begin{cases} 1 & \text{if } n = 0 \\ n \times (n-1)! & \text{otherwise} \end{cases}$

Factorial

$n!$ (n factorial)

Recursive definition: $n! = \begin{cases} 1 & \text{if } n = 0 \\ n \times (n-1)! & \text{otherwise} \end{cases}$

Combinatorial definition: $n!$ is the **number of permutations** of $1 \ldots n$

Combinations

Definition

Let n and r be nonnegative integers. Define $\binom{n}{r}$ (n choose r, also C_r^n) to be the number of ways to choose r elements from $1 \ldots n$.

Example: $\binom{4}{2} = 6$

Combinations

Definition

Let n and r be nonnegative integers. Define $\binom{n}{r}$ (n choose r, also C_r^n) to be the number of ways to choose r elements from $1 \ldots n$.

Example: $\binom{4}{2} = 6$

Combinations:

1


2


3


4


5


6


Combinations

Definition

Let n and r be nonnegative integers. Define $\binom{n}{r}$ (n choose r, also C_r^n) to be the number of ways to choose r elements from $1 \ldots n$.

Boundary Cases:

- $\binom{n}{0} = 1$
- $\binom{n}{n} = 1$
- $\binom{n}{r} = 0$ for $r > n$.
- Sometimes we extend the definition and say $\binom{n}{r} = 0$ for negative r.

Explicit Formula for $\binom{n}{r}$

Theorem

For $0 \leq r \leq n$, the following holds:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

What are we counting?

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

What are we counting?

The number of permutations of $1 \dots n$.

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

What are we counting?

The number of permutations of $1 \dots n$.

- LHS: By definition, it is $n!$.

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

What are we counting?

The number of permutations of $1 \dots n$.

- LHS: By definition, it is $n!$.
- RHS: Choose r elements out of n elements to be the first r elements.

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

What are we counting?

The number of permutations of $1 \dots n$.

- LHS: By definition, it is $n!$.
- RHS: Choose r elements out of n elements to be the first r elements.
  - For the first r elements, there are $r!$ ways to permute them.
  - For the next $n-r$ elements, there are $(n-r)!$ ways to permute them.

Explicit Formula for $\binom{n}{r}$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{1}$$

We show that $n! = \binom{n}{r} r!(n-r)!$.

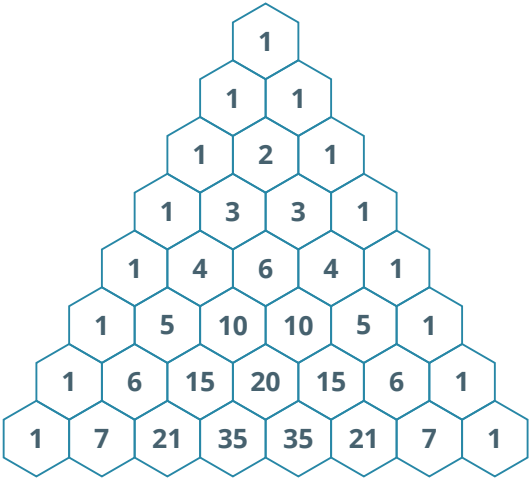
What are we counting?

The number of permutations of $1 \dots n$.

- LHS: By definition, it is $n!$.
- RHS: Choose r elements out of n elements to be the first r elements.
  - For the first r elements, there are $r!$ ways to permute them.
  - For the next $n-r$ elements, there are $(n-r)!$ ways to permute them.

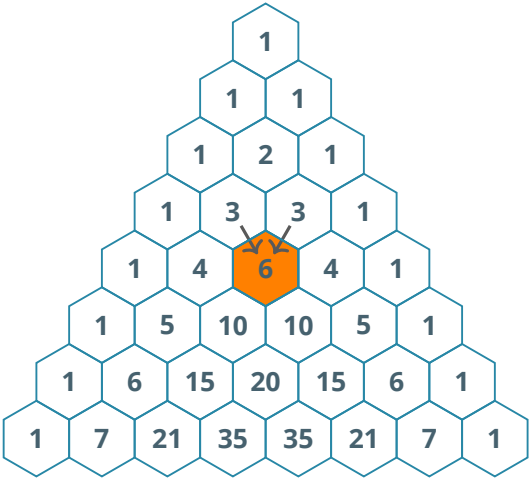
  By multiplication principle, there are $\binom{n}{r} r!(n-r)!$ permutations.

Combinations and Pascal's Triangle



Pascal Triangle: The r-th column on the n-th row is $\binom{n}{k}$.

Combinations and Pascal's Triangle

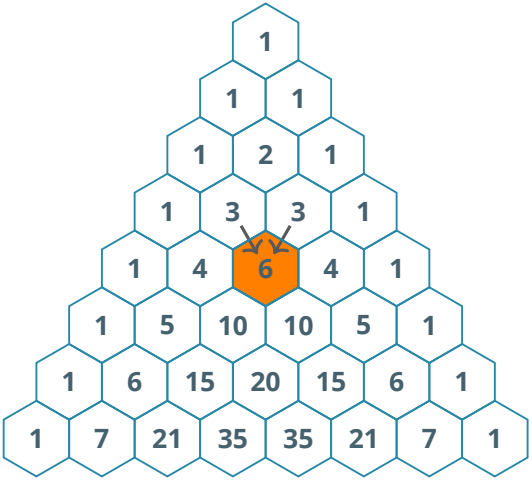


Pascal Triangle: The r-th column on the n-th row is $\binom{n}{k}$.

Pascal's Identity:

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$$

Combinations and Pascal's Triangle



Pascal Triangle: The r-th column on the n-th row is $\binom{n}{k}$.

Pascal's Identity:

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$$

Proof: Go to Math in OI (II) lecture!

Computing $\binom{n}{r} \bmod M, n \leq 2000$

Use the formula

$$\binom{n}{r} \equiv \binom{n-1}{r} + \binom{n-1}{r-1} \mod M.$$

Code:

```
for (int i = 0; i <= 2000; i++){
    for (int j = 0; j <= i; j++){
        if (j == 0 || j == i)
            ncr[i][j] = 1 % M;
        else
            ncr[i][j] = (ncr[i - 1][j] + ncr[i - 1][j - 1]) % M;
    }
}
```

$O(n^2)$ precomputation, $O(1)$ per query.

Computing $\binom{n}{r}$ mod $10^9 + 7$, $n \leq 10^6$

- $10^9 + 7$ can be replaced by any fixed large prime P.
- Precompute `fact[k]` $:= k!$ mod P and `inv_fact[k]` $:= (k!)^{-1}$ mod P.
- Output `fact[n] * inv_fact[r] * inv_fact[n - r]` mod P.

$O(n)$ precomputation, $O(1)$ per query.

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 1$ for $i = 1, 2, 3$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 1$ for $i = 1, 2, 3$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 1$ for $i = 1, 2, 3$?

Answer: $\binom{9}{2}$

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 1$ for $i = 1, 2, 3$?

Answer: $\binom{9}{2}$

- What is the number of solutions to $x_1 + x_2 + \ldots + x_r = n$, where $x_i \geq 1$ for $i = 1, 2, \ldots, r$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 1$ for $i = 1, 2, 3$?

Answer: $\binom{9}{2}$

- What is the number of solutions to $x_1 + x_2 + \ldots + x_r = n$, where $x_i \geq 1$ for $i = 1, 2, \ldots, r$?

Answer: $\binom{n-1}{r-1}$

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Let $y_i = x_i + 1$, then $y_1 + y_2 + y_3 = 13$, $y_i \geq 1$.

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Let $y_i = x_i + 1$, then $y_1 + y_2 + y_3 = 13$, $y_i \geq 1$. Answer: $\binom{12}{2}$

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Let $y_i = x_i + 1$, then $y_1 + y_2 + y_3 = 13$, $y_i \geq 1$. Answer: $\binom{12}{2}$

- What is the number of solutions to $x_1 + x_2 + \ldots + x_r = n$, where $x_i \geq 0$ for $i = 1, 2, \ldots, r$?

Counting Techniques – Stars and Bars

- What is the number of solutions to $x_1 + x_2 + x_3 = 10$, where $x_i \geq 0$ for $i = 1, 2, 3$?

Let $y_i = x_i + 1$, then $y_1 + y_2 + y_3 = 13$, $y_i \geq 1$. Answer: $\binom{12}{2}$

- What is the number of solutions to $x_1 + x_2 + \ldots + x_r = n$, where $x_i \geq 0$ for $i = 1, 2, \ldots, r$?
  Answer: $\binom{r+n-1}{r}$

Table of Contents

Set Union, Set Intersection

Definition

Given two sets A and B.
Their union $A \cup B$ is the set of elements belonging to *at least one* of A and B.
Their intersection $A \cap B$ is the set of elements belonging to *both* of A and B.

Example 1
$A := \{\text{Alex}, \text{Anson}, \text{Jason}\}, B := \{\text{Alex}, \text{Sampson}, \text{RB}\}$.
Then $A \cup B = \{\text{Alex}, \text{Anson}, \text{Jason}, \text{Sampson}, \text{RB}\}$; $A \cap B = \{\text{Alex}\}$.

Example 2
$A := \{\text{prime numbers}\}, B := \{\text{even numbers}\}$
Then $A \cap B = \{2\}$.

Mental Shortcut: intersection = AND, union = OR

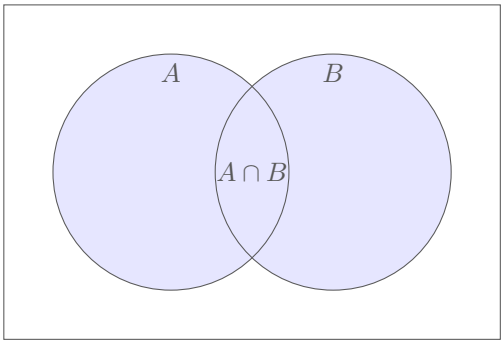
Subset, Complement

Definition

Given two sets X and Y.
Y is said to be a subset of X if all elements of Y can be found in X.
We write $Y \subseteq X$.
If $Y \subseteq X$, define its complement (with respect to X) $X \backslash Y$ to be the set of elements in X but not in Y.
We also use Y^c to denote $X \backslash Y$ when it is clear what X is.

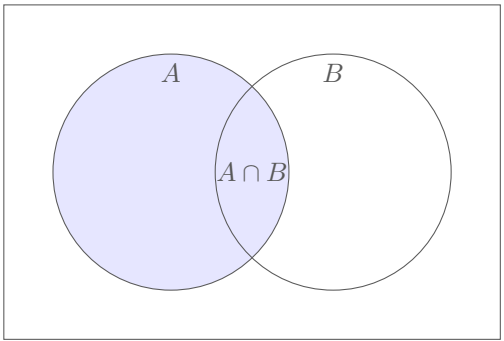
Example
Let $X :=$ {all positive integers}, $Y := \{1, 4, 9, 16, \dots\}$.
Then $Y \subseteq X$ and $X \backslash Y =$ {positive integers that are not perfect squares}.

Inclusion-Exclusion (for two sets)



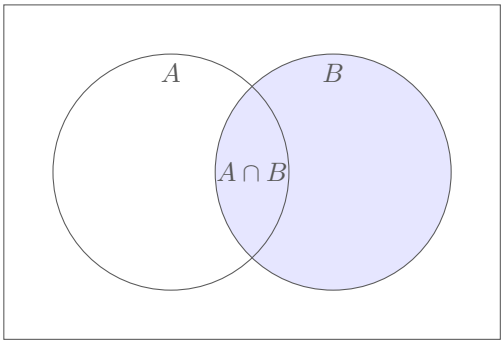
Generic form:
$|A \cup B|$

Inclusion-Exclusion (for two sets)



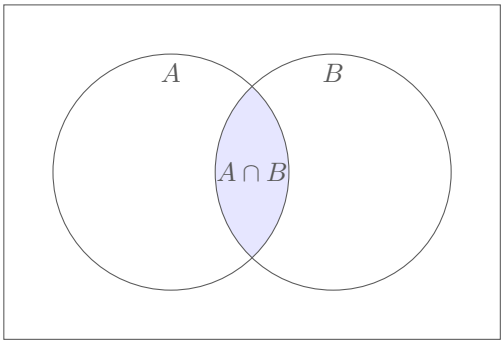
Generic form:
$|A \cup B| = |A|$

Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B|$

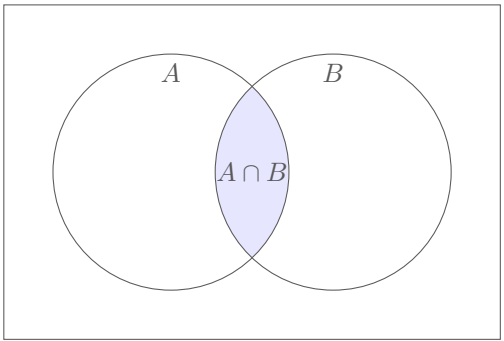
Inclusion-Exclusion (for two sets)



Generic form:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

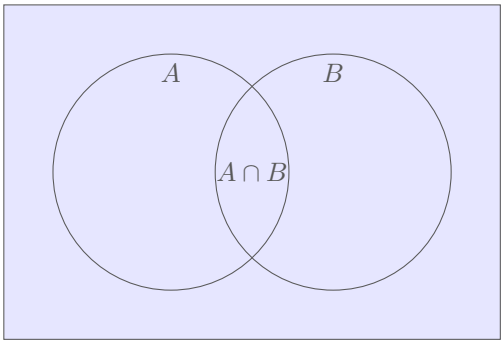
Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B| - |A \cap B|$

Alternative form:
$|A \cap B|$

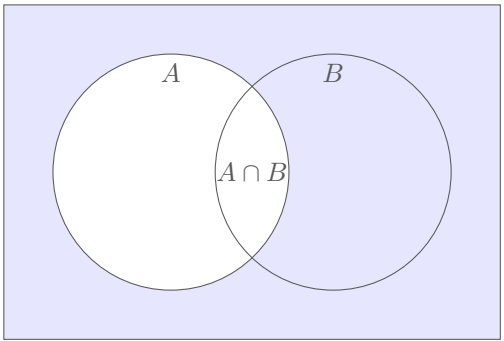
Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B| - |A \cap B|$

Alternative form:
$|A \cap B| = |X|$

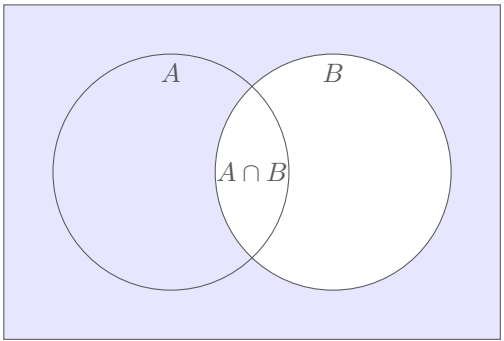
Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B| - |A \cap B|$

Alternative form:
$|A \cap B| = |X| - |A^c|$

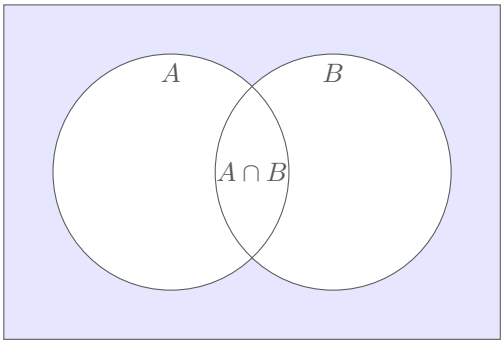
Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B| - |A \cap B|$

Alternative form:
$|A \cap B| = |X| - |A^c| - |B^c|$

Inclusion-Exclusion (for two sets)



Generic form:
$|A \cup B| = |A| + |B| - |A \cap B|$

Alternative form:
$|A \cap B| = |X| - |A^c| - |B^c| + |A^c \cap B^c|$

Inclusion-Exclusion (for two sets)

Generic form:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

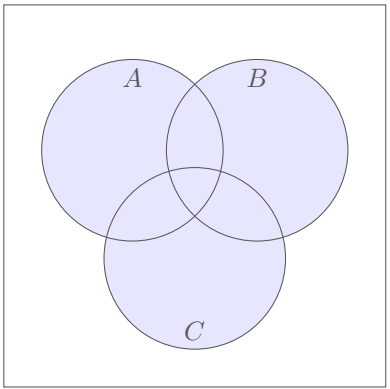
- Answers "how many elements satisfy at least one of the conditions?"
- Useful when **satisfaction of specific conditions** is easier to count

Alternative form:

$$|A \cap B| = |X| - |A^c| - |B^c| + |A^c \cap B^c|$$

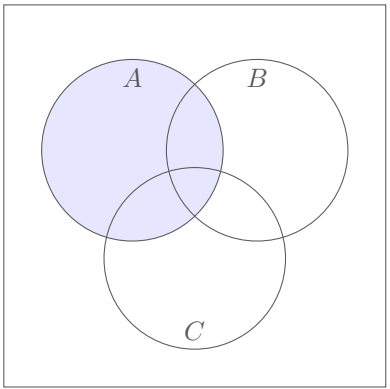
- Answers "how many elements satisfy both conditions?"
- Useful when **violation of specific conditions** is easier to count

Inclusion-Exclusion (for three sets)



Generic form:
$|A \cup B \cup C|$

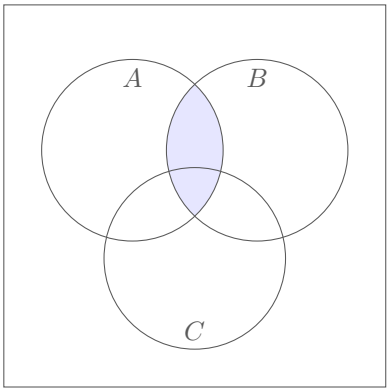
Inclusion-Exclusion (for three sets)



Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C|$$

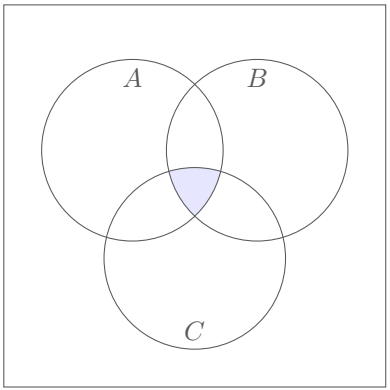
Inclusion-Exclusion (for three sets)



Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C|$$

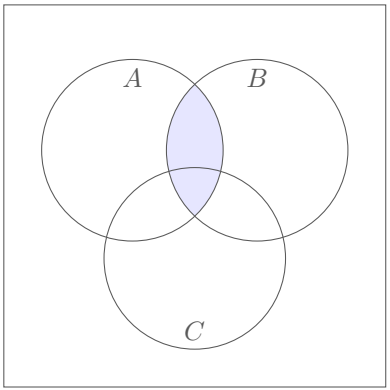
Inclusion-Exclusion (for three sets)



Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C|$$
$$-|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Inclusion-Exclusion (for three sets)



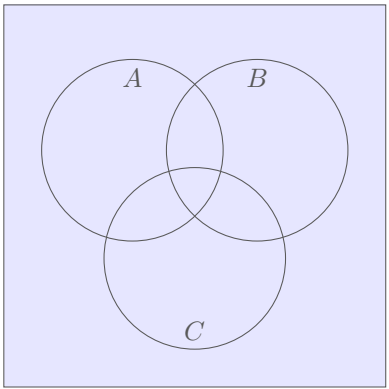
Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Alternative form:

$$|A \cap B|$$

Inclusion-Exclusion (for three sets)



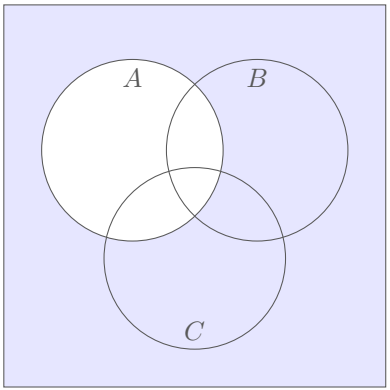
Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Alternative form:

$$|A \cap B| = |X|$$

Inclusion-Exclusion (for three sets)



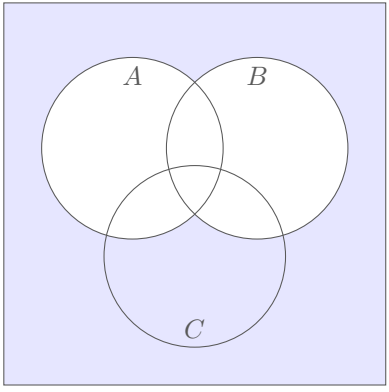
Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Alternative form:

$$|A \cap B| = |X| - |A^c| - |B^c| - |C^c|$$

Inclusion-Exclusion (for three sets)



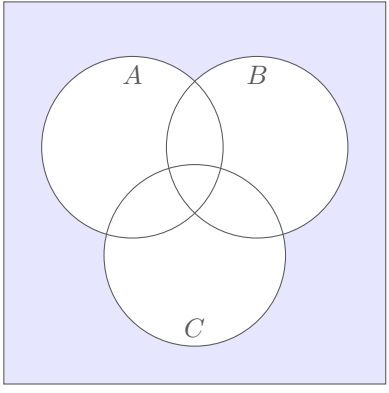
Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Alternative form:

$$|A \cap B| = |X| - |A^c| - |B^c| - |C^c| \\ +|A^c \cap B^c| + |A^c \cap C^c| + |B^c \cap C^c|$$

Inclusion-Exclusion (for three sets)



Generic form:

$$|A \cup B \cup C| = |A| + |B| + |C| \\ -|A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Alternative form:

$$|A \cap B| = |X| - |A^c| - |B^c| - |C^c| \\ +|A^c \cap B^c| + |A^c \cap C^c| + |B^c \cap C^c| \\ +|A^c \cap B^c \cap C^c|$$

Inclusion-Exclusion (for three sets)

Generic form:

$$\begin{aligned} |A \cup B \cup C| &= |A| + |B| + |C| \\ &\quad -(|A \cap B| + |A \cap C| + |B \cap C|) \\ &\quad +|A \cap B \cap C| \end{aligned}$$

Alternative form:

$$\begin{aligned} |A \cap B \cap C| &= |X| \\ &\quad -(|A^c| + |B^c| + |C^c|) \\ &\quad +(|A^c \cap B^c| + |A^c \cap C^c| + |B^c \cap C^c|) \\ &\quad -|A^c \cap B^c \cap C^c| \end{aligned}$$

Can you see a pattern?

Inclusion-Exclusion (for n sets)

Given $S_1, \ldots, S_n \subseteq X$.
Generic form:

$$|S_1 \cup \cdots \cup S_n| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \le i_1 < \cdots < i_k \le n} |S_{i_1} \cap \cdots \cap S_{i_k}|$$

Alternative form:

$$|S_1 \cap \cdots \cap S_n| = \sum_{k=0}^{n} (-1)^{k} \sum_{1 \le i_1 < \cdots < i_k \le n} |S_{i_1}^c \cap \cdots \cap S_{i_k}^c|$$

Solving M1832

Problem statement: find the number of integral solutions to $x_1 + \cdots + x_n = T$, subject to $0 \leq x_i \leq a_i$. Output the answer modulo $10^9 + 7$.

Constraints: $1 \leq n \leq 16, 1 \leq T \leq 10^9, 1 \leq a_i \leq 10^9$.

Solving M1832

Problem statement: find the number of integral solutions to $x_1 + \cdots + x_n = T$, subject to $0 \leq x_i \leq a_i$. Output the answer modulo $10^9 + 7$.

Constraints: $1 \leq n \leq 16, 1 \leq T \leq 10^9, 1 \leq a_i \leq 10^9$.

Solution Idea:

- Let X be the solution set to $x_1 + \cdots + x_n = S$, without upper bound constraints.
- Let S_i consist of elements of X with $x_i \leq a_i$.
- We want to find $|S_1 \cap S_2 \cap \cdots \cap S_n|$.
- Handling constraint violation is easy! If $x_i > a_i$ we just use $x'_i := x_i - a_i - 1$ to get back an unconstrained equation. So we know how to calculate quantities like $|S^c_{i_1} \cap \cdots \cap S^c_{i_k}|$ quickly.
- Use Inclusion-Exclusion to find the answer.

Reference

- Wikipedia
- Past Mathematics in OI (I), (II) sides - 2015 - 2024

Practice Problems

HKOJ 20374 Big Mod
HKOJ I0501 Divisor Game (Interactive)
HKOJ M0723 Frog
HKOJ J041 Traffic Lights

HKOJ 01029 N Collinear Planets
HKOJ M1631 A Strange Elevator
HKOJ M1821 Contest Score
HKOJ M1822 Power Tower

More:
Codeforces - Number Theory Tag
Codeforces - CRT Tag

Questions?