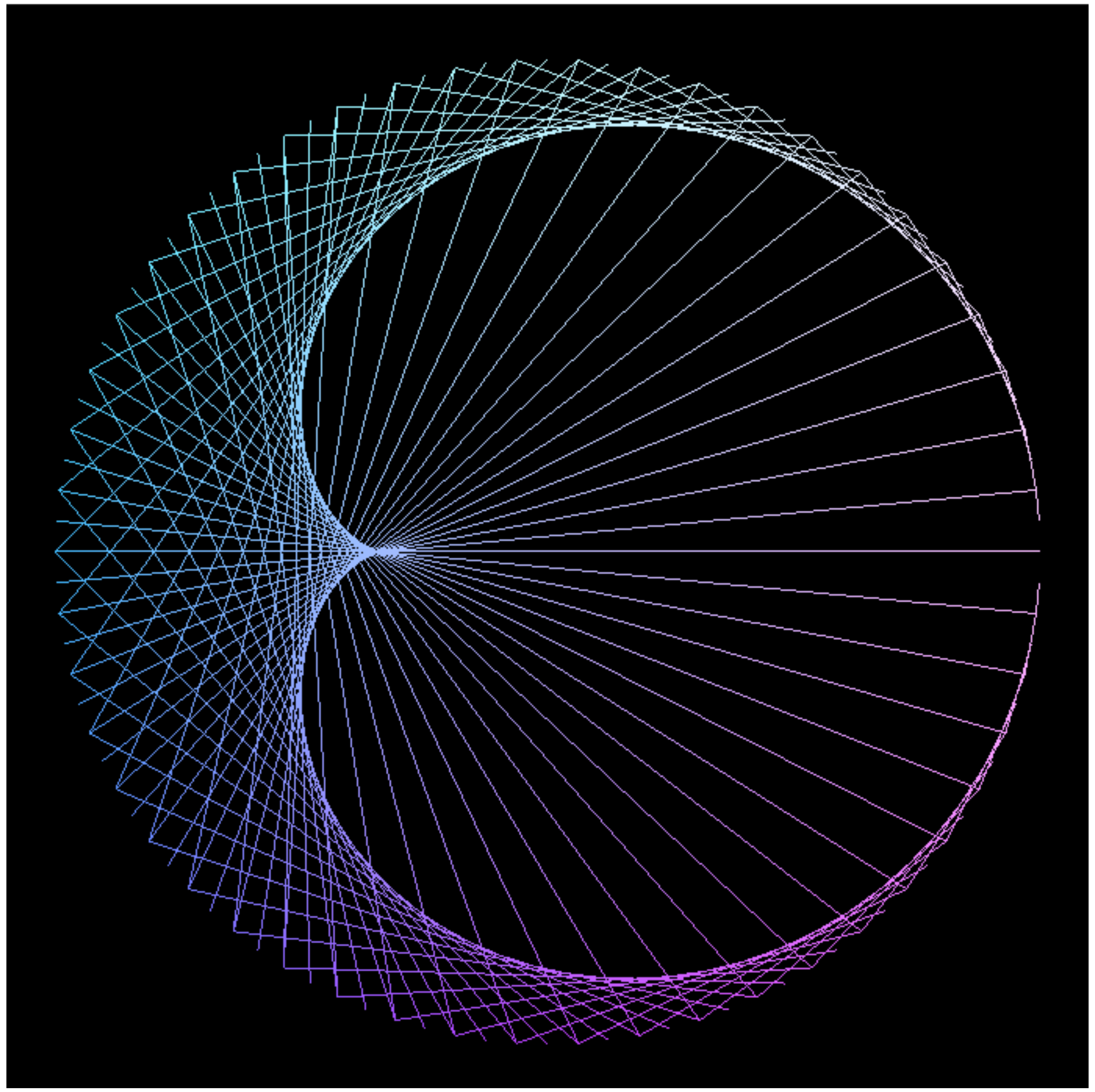
CS251

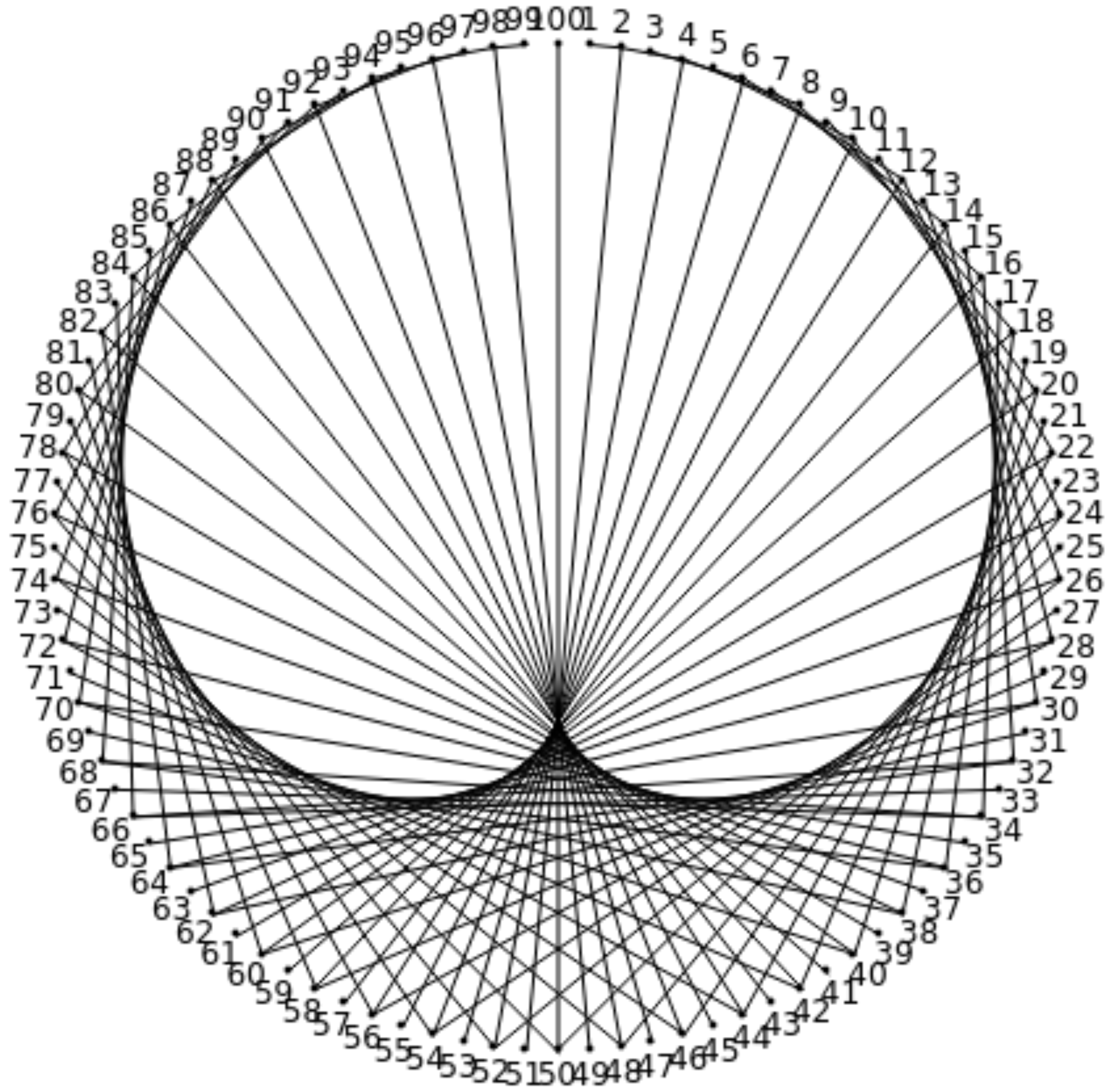
Great Ideas
in
Theoretical
Computer Science



Modular Arithmetic

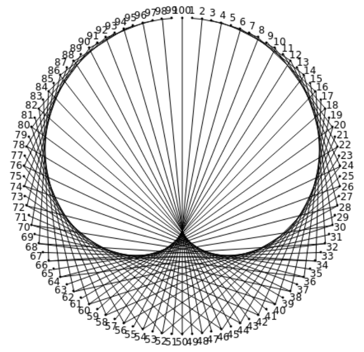
This module

Modular Arithmetic



Cryptography

Goal of this lecture



Understand **modular arithmetic**: **Theory** + **Algorithms**

Why:

1. infinite universe: what we are used to (e.g. $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$)

   finite universe: what we sometimes prefer

2. Some hard-to-do arithmetic operations in $\mathbb{Z}$ or $\mathbb{Q}$

   are easy in the modular universe.

3. Some easy-to-do arithmetic operations in $\mathbb{Z}$ or $\mathbb{Q}$

   seem to be hard in the modular universe.

   And this is great for cryptography applications!

Goal of this lecture

Understand **modular arithmetic**: **Theory** + **Algorithms**

- How to view elements of the **modular universe**?

- How to do basic operations in the **modular universe**:

1. addition
2. subtraction
3. multiplication
4. division
5. exponentiation
6. taking roots
7. logarithm

Theory (definitions)

+

Algorithms

efficient (?)

The plan

Start with **good old integers**.

Move to **modular universe**.

poll.cs251.com

Wilson's Theorem: N is prime **iff** $(N-1)! \equiv N-1 \mod N$.

Can we use this (in the obvious way) to design a poly-time algorithm for isPrime?

Input: $A, E \in \mathbb{Z}^+$

Output: $\lceil A^{1/E} \rceil$

Thoughts?

Algorithm:

Linear search: Try $B = 1, 2, 3, \ \dots$ Stop when $B^E \geq A$.

Input: $A, B \in \mathbb{Z}^+$

Output: $\lceil \log_B A \rceil$

Thoughts?

Algorithm:

Linear search: Try $E = 1, 2, 3, \ \dots$ Stop when $B^E \geq A$.

Integer Universe

Algorithms on numbers involve **BIG** numbers.

3618502788666131106986593281521497110455743021169260358536775932020762686101
7237846234873269807102970128874356021481964232857782295671675021393065473695
3943653222082116941587830769649826310589717739181525033220266350650989268038
3194839273881505432422077179121838888281996148408052302196889866637200606252
6501310964926475205090003984176122058711164567946559044971683604424076996342
7183046544798021168297013490774140090476348290671822743961203698142307099664
3455133414637616824423860107889741058131271306226214208636008224651510961018
9789006815067664901594246966730927620844732714004599013904409378141724958467
7228950143608277369974692883195684314361862929679227167524851316077587207648
7845058367231603173079817471417519051357029671991152963580412838184841733782

Integer Universe

$n = \text{len}(B) =$ # bits to write B

$$n = \text{len}(B) \approx \log_2 B \quad \Longrightarrow \quad B \approx 2^n.$$

Example:

$$B = 5693030020523999993479642904621911725098567020556258102766251487234031094429$$

$B \approx 5.7 \times 10^{75}$ (# particles in the universe)

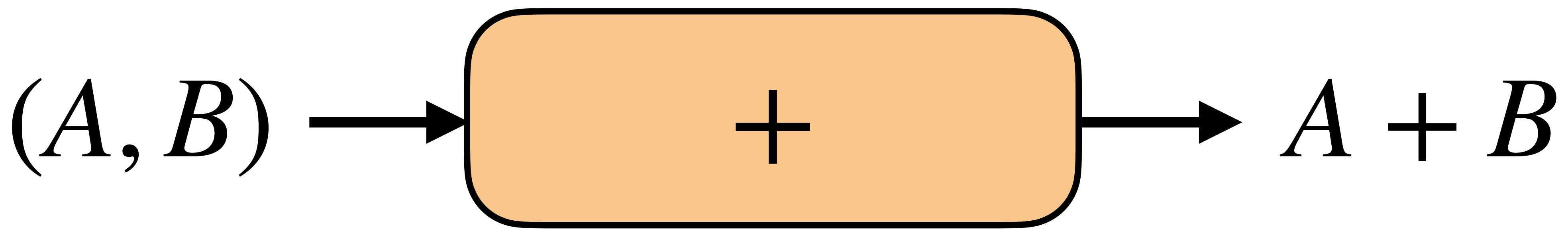
$$n = \text{len}(B) = 251$$

An algorithm repeating B times is **practically uncomputable**.

Integer Universe: Addition complexity

Input: $A, B \in \mathbb{Z}$

Output: $A + B$



$$(A, B) \longrightarrow \boxed{+} \longrightarrow A + B$$

Algorithm:

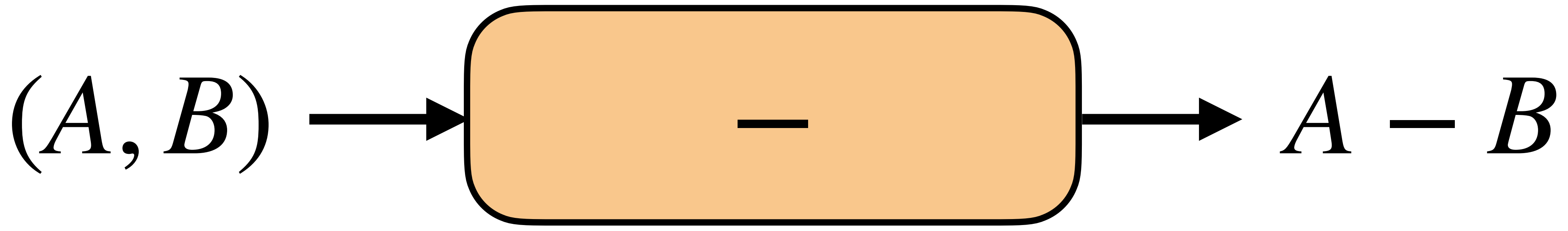
Grade school algorithm.

Complexity: Poly-time

Integer Universe: Subtraction complexity

Input: $A, B \in \mathbb{Z}$

Output: $A - B$



$(A, B) \rightarrow \boxed{-} \rightarrow A - B$

Algorithm:

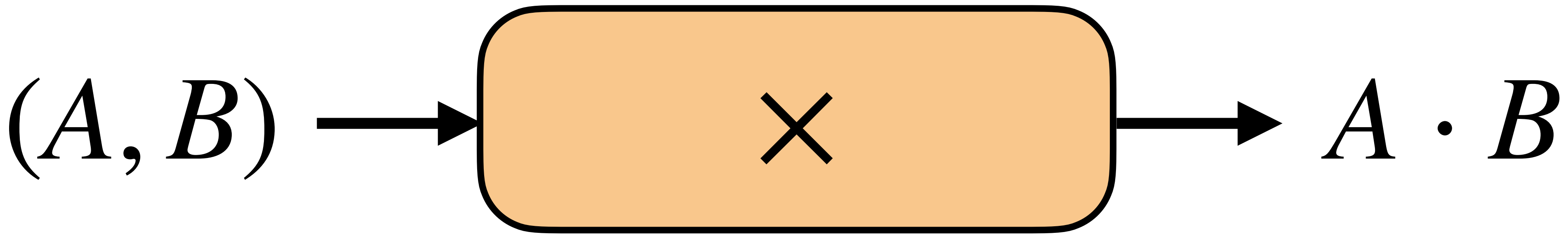
Grade school algorithm.

Complexity: Poly-time

Integer Universe: Multiplication complexity

Input: $A, B \in \mathbb{Z}$

Output: $A \cdot B$



Algorithm:

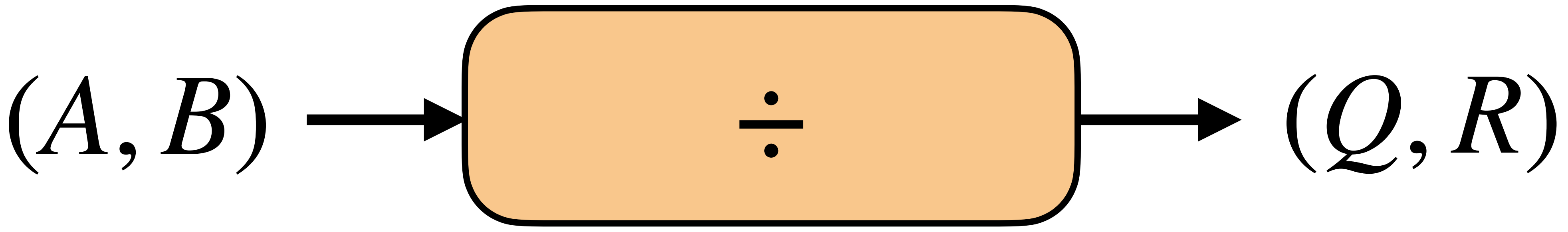
- Grade school long multiplication.
- Karatsuba.
- ...

Complexity: Poly-time

Integer Universe: Division complexity

Input: $A, B \in \mathbb{Z}$

Output: (Q, R) where $R = A \bmod B$ and $A = Q \cdot B + R$



Algorithm:

- Grade school long division.
- …

Complexity: Poly-time

Integer Universe: Exponentiation complexity

Input: $B, E \in \mathbb{Z}$

Output: B^E

$$(B, E) \longrightarrow \boxed{\text{EXP}} \longrightarrow B^E$$

Let $B = 2$ and $E = 569303002052399999347964290462191172509856702055625810276625148723403118357992154094429925775349 58757$

Input length: $\text{len}(E) = \log(E).$

Output length: $\text{len}(B^E) = \log(B^E) = E.$ (exceeds # particles in the universe)

Complexity: Exponential time.

Integer Universe: Logarithm complexity

Input: $A, B \in \mathbb{Z}^+$

Output: $\log_B A$ (i.e. E such that $B^E = A$)

$$(A = B^E, B) \longrightarrow \boxed{\mathrm{LOG}_B} \longrightarrow E$$

Algorithm:

- Linear search. Try $E = 1, 2, 3, \ldots$

  Stop when $B^E \geq A$.

Complexity: Poly-time

Integer Universe: Root complexity

Input: $A, E \in \mathbb{Z}^+$

Output: $A^{1/E}$ (i.e. B such that $B^E = A$)

$$(A = B^E, E) \longrightarrow \boxed{\mathrm{ROOT}_E} \longrightarrow B$$

Algorithm:

~~- Linear search.~~

- Binary search.

Complexity: Poly-time

The plan

Start with **good old integers**. ✓

Move to **modular universe**.

Goal of this lecture

Understand **modular arithmetic**: **Theory** + **Algorithms**

- How to view elements of the **modular universe**?

- How to do basic operations in the **modular universe**:

1. addition

2. subtraction

3. multiplication

4. division

5. exponentiation

6. taking roots

7. logarithm

Theory (definitions)

+

Algorithms

efficient (?)

Modular Universe: How to view the elements

$A \text{ mod } N:$ remainder when you divide A by N.

Example $N = 5$

$$\begin{tikzcd}
0 \arrow[d] & 1 \arrow[d] & 2 \arrow[d] & 3 \arrow[d] & 4 \arrow[d] & 5 \arrow[d] & 6 \arrow[d] & 7 \arrow[d] & 8 \arrow[d] & 9 \arrow[d] & 10 \arrow[d] & 11 \arrow[d, "\text{mod } 5"] & \cdots \\
0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & \cdots
\end{tikzcd}$$

Modular Universe: How to view the elements

$A \bmod N$: remainder when you divide A by N.

Notation: $A \equiv B \mod N$ or $A \equiv_N B$

"A is congruent to B modulo N"

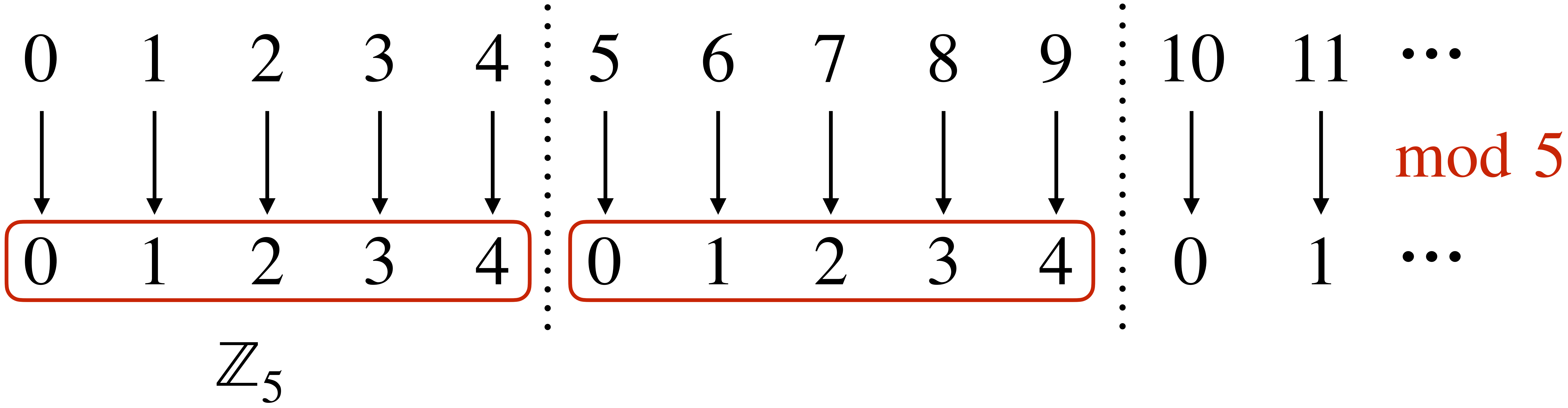
$(A \bmod N) = (B \bmod N)$

Modular Universe: How to view the elements

2 Points of View

View 1 The universe is $\mathbb{Z}$.

Every element has a $\text{mod } N$ representation.



0	1	2	3	4	5	6	7	8	9	10	11	$\cdots$
0	1	2	3	4	0	1	2	3	4	0	1	$\cdots$

$\text{mod } 5$

$\mathbb{Z}_5$

View 2 The universe is the finite set $\mathbb{Z}_N = \{0, 1, 2, \ldots, N - 1\}$.

Goal of this lecture

Understand **modular arithmetic**: **Theory** + **Algorithms**

- How to view elements of the **modular universe**?

- How to do basic operations in the **modular universe**:

1. addition
2. subtraction
3. multiplication
4. division
5. exponentiation
6. taking roots
7. logarithm

Theory (definitions)

+

Algorithms

efficient (?)

Modular Universe: Addition

Can define a "plus" operation for the universe $\mathbb{Z}_N$.

For $A, B \in \mathbb{Z}_N$:

$$A \underbrace{+_N}_{\boxed{\text{"plus" in } \mathbb{Z}_N}} B \overset{\text{def}}{=} (A \underbrace{+}_{\boxed{\text{plus in } \mathbb{Z}}} B) \bmod N$$

Modular Universe: Addition

Addition table for $\mathbb{Z}_5$

$+_N$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

0 is the (additive) **identity**:

$0 +_N A = A +_N 0 = A$ for any A.

Modular Universe: Addition

In $\mathbb{Z}$		In $\mathbb{Z}_5$
3019573	$\xrightarrow{\text{mod } 5}$	3
912382237	$\xrightarrow{\text{mod } 5}$	2
3019573 + 912382237	$\xrightarrow{\text{mod } 5}$	0 **?** **YES!**

Modular Universe: Addition

In $\mathbb{Z}$		In $\mathbb{Z}_N$
A	$\xrightarrow{\text{mod } N}$	$A \bmod N$
B	$\xrightarrow{\text{mod } N}$	$B \bmod N$
$A + B$	$\xrightarrow{\text{mod } N}$	$(A \bmod N) +_N (B \bmod N)$ **?**

YES!

Modular Universe: Subtraction

What does $A - B$ mean?

Addition in disguise: $A + (-B)$

What does $-B$ mean?

Definition: The **additive inverse** of $B \in \mathbb{Z}_N$, denoted $-B$, is the element in $\mathbb{Z}_N$ such that $B +_N -B = 0$.

$$A -_N B \overset{\text{def}}{=} A +_N -B$$

Modular Universe: Subtraction

Addition table for $\mathbb{Z}_5$

$+_N$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

<u>additive inverses</u>

$-0 = 0$

$-1 = 4$

$-2 = 3$

$-3 = 2$

$-4 = 1$

Modular Universe: Subtraction

Addition table for $\mathbb{Z}_5$

$+_N$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Fact:

For every $A \in \mathbb{Z}_N$, $-A$ exists. (why?)

Corollary:

Each row contains distinct elements.

i.e. every row is a permutation of $\mathbb{Z}_N$.

Modular Universe: Multiplication

Can define a "multiplication" operation for the universe $\mathbb{Z}_N$.

For $A, B \in \mathbb{Z}_N$: $$A \cdot_N B \overset{\text{def}}{=} (A \cdot B) \bmod N$$

"multiplication" in $\mathbb{Z}_N$

multiplication in $\mathbb{Z}$

Modular Universe: Multiplication

Multiplication table for $\mathbb{Z}_5$

$\cdot_N$	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

1 is the (multiplicative) **identity**:

$1 \ \cdot_N A = A \cdot_N 1 = A$ for any A

Modular Universe: Multiplication

In $\mathbb{Z}$		In $\mathbb{Z}_N$
A	$\xrightarrow{\text{mod } N}$	$A \bmod N$
B	$\xrightarrow{\text{mod } N}$	$B \bmod N$
$A \cdot B$	$\xrightarrow{\text{mod } N}$	$(A \bmod N) \cdot_N (B \bmod N)$ **?**

YES!

Modular Universe: Division

What does A/B mean?

Multiplication in disguise: $A \cdot \frac{1}{B} = A \cdot B^{-1}$

What does B^{-1} mean?

Definition: The **multiplicative inverse** of $B \in \mathbb{Z}_N$, denoted B^{-1}, is the element in $\mathbb{Z}_N$ such that $B \cdot_N B^{-1} = 1$.

$$A /_N B \overset{\text{def}}{=} A \cdot_N B^{-1}$$

Modular Universe: Division

Multiplication table for $\mathbb{Z}_5$

$\cdot_N$	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

multiplicative inverses

0^{-1} = undefined

$1^{-1} = 1$

$2^{-1} = 3$

$3^{-1} = 2$

$4^{-1} = 4$

Modular Universe: Division

Multiplication table for $\mathbb{Z}_6$

$\cdot_N$	**0**	**1**	**2**	**3**	**4**	**5**
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

multiplicative inverses

0^{-1} = undefined

$1^{-1} = 1$

2^{-1} = undefined

3^{-1} = undefined

4^{-1} = undefined

$5^{-1} = 5$

WTF?

Modular Universe: Division

Multiplication table for $\mathbb{Z}_7$

$\cdot_N$	**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**

Every element except 0
has a multiplicative inverse.

Modular Universe: Division

Multiplication table for $\mathbb{Z}_8$

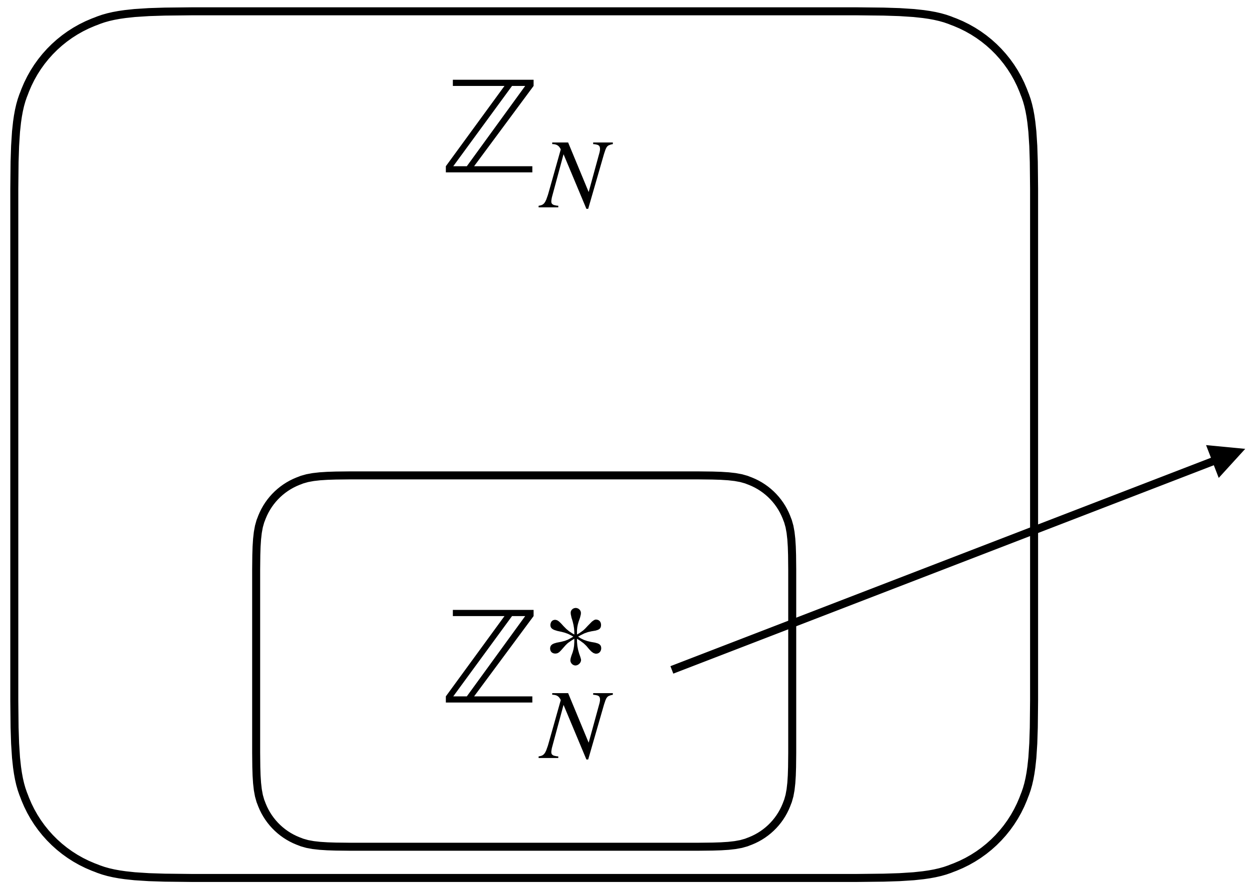
$\cdot_N$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

1, 3, 5, 7 have inverses.

Others don't.

Modular Universe: Division

Fact: $A^{-1} \in \mathbb{Z}_N$ exists **iff** $\gcd(A, N) = 1.$



$\{A \in \mathbb{Z}_N : \gcd(A, N) = 1\}$

Notation: $\varphi(N) = |\mathbb{Z}_N^*|$

Is $\mathbb{Z}_N^*$ is "closed" under multiplication?

i.e. $A, B \in \mathbb{Z}_N^* \implies A \cdot_N B \in \mathbb{Z}_N^*$?

Modular Universe: Division

$\mathbb{Z}_5^*$

$\cdot_N$	~~**0**~~	**1**	**2**	**3**	**4**
~~**0**~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
1	~~0~~	1	2	3	4
2	~~0~~	2	4	1	3
3	~~0~~	3	1	4	2
4	~~0~~	4	3	2	1

$$\varphi(5) = 4$$

Modular Universe: Division

$\mathbb{Z}_5^*$

$\cdot_N$	**1**	**2**	**3**	**4**
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\varphi(5) = 4$

N For P prime, $\varphi(P) = P - 1$.

Modular Universe: Division

$\mathbb{Z}_8^*$

$\cdot_N$	~~**0**~~	**1**	~~**2**~~	**3**	~~**4**~~	**5**	~~**6**~~	**7**
~~**0**~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
1	~~0~~	1	~~2~~	3	~~4~~	5	~~6~~	7
~~**2**~~	~~0~~	~~2~~	~~4~~	~~6~~	~~0~~	~~2~~	~~4~~	~~6~~
3	~~0~~	3	~~6~~	1	~~4~~	7	~~2~~	5
~~**4**~~	~~0~~	~~4~~	~~0~~	~~4~~	~~0~~	~~4~~	~~0~~	~~4~~
5	~~0~~	5	~~2~~	7	~~4~~	1	~~6~~	3
~~**6**~~	~~0~~	~~6~~	~~4~~	~~2~~	~~0~~	~~6~~	~~4~~	~~2~~
7	~~0~~	7	~~6~~	5	~~4~~	3	~~2~~	1

$\varphi(8) = 4$

Modular Universe: Division

$$\mathbb{Z}_8^*$$

$\cdot_N$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$$\varphi(8) = 4$$

Modular Universe: Division

$\mathbb{Z}_{15}^*$

$\cdot_N$	**1**	**2**	**4**	**7**	**8**	**11**	**13**	**14**
1	1	2	4	7	8	11	13	14
2	2	4	8	14	1	7	11	13
4	4	8	1	13	2	14	7	11
7	7	14	13	4	11	2	1	8
8	8	1	2	11	4	13	14	7
11	11	7	14	2	13	1	8	4
13	13	11	7	1	14	8	4	2
14	14	13	11	8	7	4	2	1

$\varphi(15) = 8$

Modular Universe: Division

$\mathbb{Z}_{15}^*$

$\cdot_N$	**1**	**2**	**4**	**7**	**8**	**11**	**13**	**14**
1	1	2	4	7	8	11	13	14
2	2	4	8	14	1	7	11	13
4	4	8	1	13	2	14	7	11
7	7	14	13	4	11	2	1	8
8	8	1	2	11	4	13	14	7
11	11	7	14	2	13	1	8	4
13	13	11	7	1	14	8	4	2
14	14	13	11	8	7	4	2	1

Exercise: For P, Q distinct primes, $\varphi(PQ) = (P-1)(Q-1)$.

Modular Universe: Division

$\mathbb{Z}_8^*$

$\cdot_N$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\varphi(8) = 4$

Fact:

For every $A \in \mathbb{Z}_N^*$, A^{-1} exists. (why?)

Corollary:

Each row contains distinct elements.

i.e. every row is a permutation of $\mathbb{Z}_N^*$.

SUMMARY

$+_N$	**0**	**1**	**2**	**3**
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\mathbb{Z}_N$

behaves nicely
with respect to
addition / subtraction

$\cdot_N$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\mathbb{Z}_N^*$

behaves nicely
with respect to
multiplication / division

Modular Universe: Exponentiation

Notation:

For $A \in \mathbb{Z}_N$ and $E \in \mathbb{N}$: $A^E = \underbrace{A \cdot_N A \cdot_N \cdots \cdot_N A}_{E \text{ times}}$

Modular Universe: Exponentiation

Notation:

For $A \in \mathbb{Z}_N^*$ and $E \in \mathbb{N}$: $$A^E = \underbrace{A \cdot_N A \cdot_N \cdots \cdot_N A}_{E \text{ times}}$$

poll.cs251.com

What is $213^{150} \bmod 7$?

Modular Universe: Exponentiation

$\mathbb{Z}_5^*$

$$\begin{array}{c|cccc} \cdot_N & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \\ \hline \mathbf{1} & 1 & 2 & 3 & 4 \\ \mathbf{2} & 2 & 4 & 1 & 3 \\ \mathbf{3} & 3 & 1 & 4 & 2 \\ \mathbf{4} & 4 & 3 & 2 & 1 \end{array}$$

$\varphi(5) = 4$

1^0	1^1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1	1
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	3	1	2	4	3	1
3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8
1	3	4	2	1	3	4	2	1
4^0	4^1	4^2	4^3	4^4	4^5	4^6	4^7	4^8
1	4	1	4	1	4	1	4	1

2 and 3 are called **generators**.

Modular Universe: Exponentiation

$\mathbb{Z}_5^*$

$\cdot_N$	**1**	**2**	**3**	**4**
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\varphi(5) = 4$

1^0	1^1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1	1
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	3	1	2	4	3	1
3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8
1	3	4	2	1	3	4	2	1
4^0	4^1	4^2	4^3	4^4	4^5	4^6	4^7	4^8
1	4	1	4	1	4	1	4	1

Modular Universe: Exponentiation

$\mathbb{Z}_8^*$

$\cdot_N$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\varphi(8) = 4$

1^0	1^1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1	1
3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8
1	3	1	3	1	3	1	3	1
5^0	5^1	5^2	5^3	5^4	5^5	5^6	5^7	5^8
1	5	1	5	1	5	1	5	1
7^0	7^1	7^2	7^3	7^4	7^5	7^6	7^7	7^8
1	7	1	7	1	7	1	7	1

Euler's Theorem: For any $A \in \mathbb{Z}_N^*$, $A^{\varphi(N)} = 1$.

Modular Universe: Exponentiation

$\mathbb{Z}_8^*$

$\cdot_N$	**1**	**3**	**5**	**7**
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\varphi(8) = 4$

Euler's Theorem: For any $A \in \mathbb{Z}_N^*$, $A^{\varphi(N)} = 1$.

Proof:

Poll Answer

What is $213^{150} \bmod 7$?

$$213^{150} \bmod 7 \quad = \quad 3^{150} \bmod 7 \quad = \quad 3^{150 \bmod 6} \bmod 7 \quad = \quad 3^{0} \bmod 7 \quad = \quad 1$$

Euler's Theorem:

$$\begin{array}{cccccccc} A^0 & A^1 & A^2 & \dots & A^{\varphi(N)} & A^{\varphi(N)+1} & A^{\varphi(N)+2} & \dots \\ & & & & \| & \| & \| & \\ & & & & A^0 & A^1 & A^2 & \dots \end{array}$$

Corollary: Can reduce the exponent mod $\varphi(N)$.

When exponentiating $A \in \mathbb{Z}_N^*$,

think of the exponent as living in the universe $\mathbb{Z}_{\varphi(N)}$.

Modular Universe: Root & Log

$\mathbb{Z}_N^*$

$$(B, E, N) \longrightarrow \boxed{\text{EXP}} \longrightarrow B^E \bmod N$$

<u>2 inverse functions:</u>

$$(B^E, E, N) \longrightarrow \boxed{\text{ROOT}_E} \longrightarrow B$$

$$(B^E, B, N) \longrightarrow \boxed{\text{LOG}_B} \longrightarrow E$$

Goal of this lecture

Understand **modular arithmetic**: **Theory** + **Algorithms**

- How to view elements of the **modular universe**?

- How to do basic operations in the **modular universe**:

1. addition
2. subtraction
3. multiplication
4. division
5. exponentiation
6. taking roots
7. logarithm

Theory (definitions)

+

Algorithms

efficient (?)

Modular Universe: Addition complexity

Input: $A, B \in \mathbb{Z}_N, \; N$

Output: $A +_N B$

$$(A, B, N) \longrightarrow \boxed{+_N} \longrightarrow A +_N B$$

Algorithm:

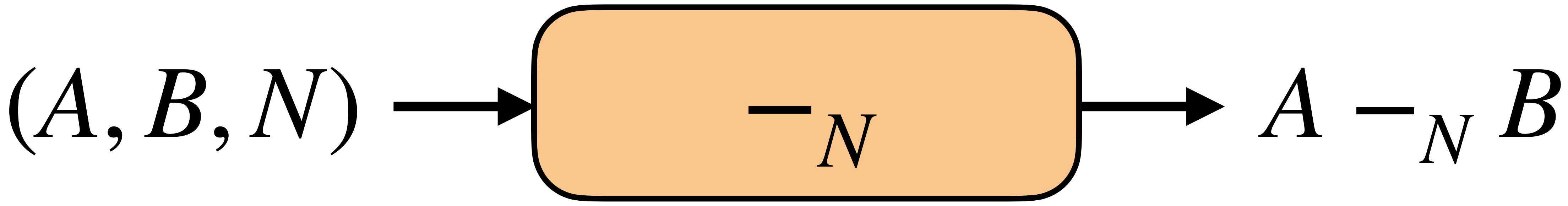
Compute $(A + B) \bmod N$.

Complexity: Poly-time

Modular Universe: Subtraction complexity

Input: $A, B \in \mathbb{Z}_N,\ N$

Output: $A -_N B$



Algorithm:

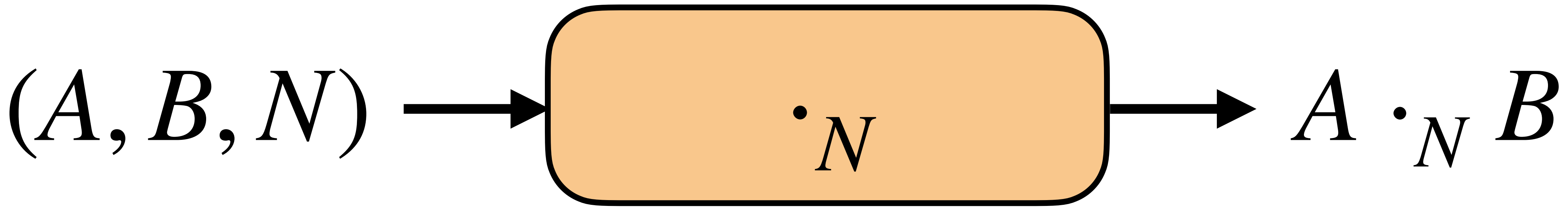
Compute $(A + (N - B)) \bmod N.$

Complexity: Poly-time

Modular Universe: Multiplication complexity

Input: $A, B \in \mathbb{Z}_N, \; N$

Output: $A \cdot_N B$



$$(A, B, N) \longrightarrow \boxed{\cdot_N} \longrightarrow A \cdot_N B$$

Algorithm:

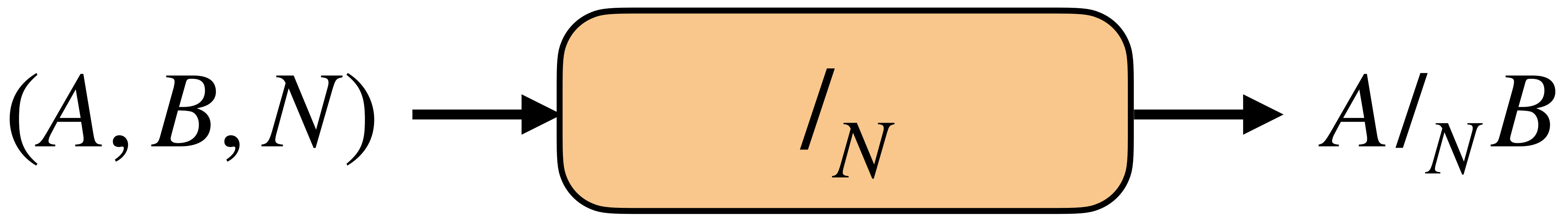
Compute $(A \cdot B) \bmod N$.

Complexity: Poly-time

Modular Universe: Division complexity

Input: $A, B \in \mathbb{Z}_N,\ N$

Output: $A /_N B$ (if exists)



$(A, B, N) \rightarrow \boxed{/_N} \rightarrow A /_N B$

Algorithm:

Compute $A \cdot_N B^{-1}$.

? Does B^{-1} exist?

? How do you compute B^{-1}?

Modular Universe: Division complexity

Does B^{-1} exist?

B^{-1} exists **iff** $\gcd(B, N) = 1$.

Euclid's alg. computes gcd in poly-time.

How do you compute B^{-1}?

Extension of Euclid's alg. gives B^{-1} in poly-time.

Modular Universe: Division complexity

Extension of Euclid's alg. gives B^{-1} in poly-time.

Definition: C is a **miix** of A and B if $C = k \cdot A + \ell \cdot B$ for some $k, \ell \in \mathbb{Z}$.

not a real term 😋

2 is a **miix** of 14 and 10: $2 = -2 \cdot 14 + 3 \cdot 10.$

7 is not a **miix** of 55 and 40. (why?)

Modular Universe: Division complexity

Extension of Euclid's alg. gives B^{-1} in poly-time.

Definition: C is a **miix** of A and B if $C = k \cdot A + \ell \cdot B$ for some $k, \ell \in \mathbb{Z}$.

not a real term 😋

Fact: $\gcd(A, B)$ is a **miix** of A and B: $\gcd(A, B) = k \cdot A + \ell \cdot B$

Exercise: Extension of Euclid's alg. spits out k and ℓ.

Finding B^{-1} modulo N:

$$\gcd(B, N) = 1 \implies \exists\, k, \ell \text{ such that } 1 = \underset{\|}{k} \cdot B + \ell \cdot N$$

Then we have found B^{-1}.

Modular Universe: Division complexity

Input: $A, B \in \mathbb{Z}_N$, N

Output: $A /_N B$ (if exists)

Algorithm:

Compute $A \cdot_N B^{-1}$.

Modular Universe: Division complexity

Input: $A, B \in \mathbb{Z}_N,\ N$

Output: $A /_N B$ (if exists)

Algorithm:

```
$(G, k, \ell) = \text{Extended-Euclid}(B, N)$        (so $G = k \cdot B + \ell \cdot N$)
if $G == 1$:
    $B^{-1} = k \bmod N$
    return $(A \cdot B^{-1}) \bmod N$
```

Complexity: Poly-time

Modular Universe: Exponentiation complexity

Input: $B \in \mathbb{Z}_N, \; E, \; N$

Output: $B^E \bmod N$

$$(B, E, N) \longrightarrow \boxed{\text{EXP}} \longrightarrow B^E \bmod N$$

N Length of output not an issue.

Algorithm:

Fast modular exponentiation. (repeatedly square & mod)

Complexity: Poly-time

Modular Universe: Log complexity

Input: $A \in \mathbb{Z}_N^*,\ B,\ N$

Output: $\log_B A$ in $\mathbb{Z}_N^*$

$$(A = B^E, B, N) \longrightarrow \boxed{\mathrm{LOG}_B} \longrightarrow E$$

No poly-time algorithm known!

Modular Universe: Root complexity

Input: $A \in \mathbb{Z}_N^*,\ E,\ N$

Output: $A^{1/E}$ in $\mathbb{Z}_N^*$

$$(A = B^E, E, N) \longrightarrow \boxed{\mathrm{ROOT}_E} \longrightarrow B$$

❗ **No poly-time algorithm known!**

Summary

	Integer Universe	Modular Universe
1. addition	✓	✓
2. subtraction	✓	✓
3. multiplication	✓	✓
4. division	✓	✓
5. exponentiation	✗	✓
6. taking roots	✓	?
7. logarithm	✓	?

Next Time

Cryptography