CS70 – Spring 2026

Lecture 7 : Feb. 10

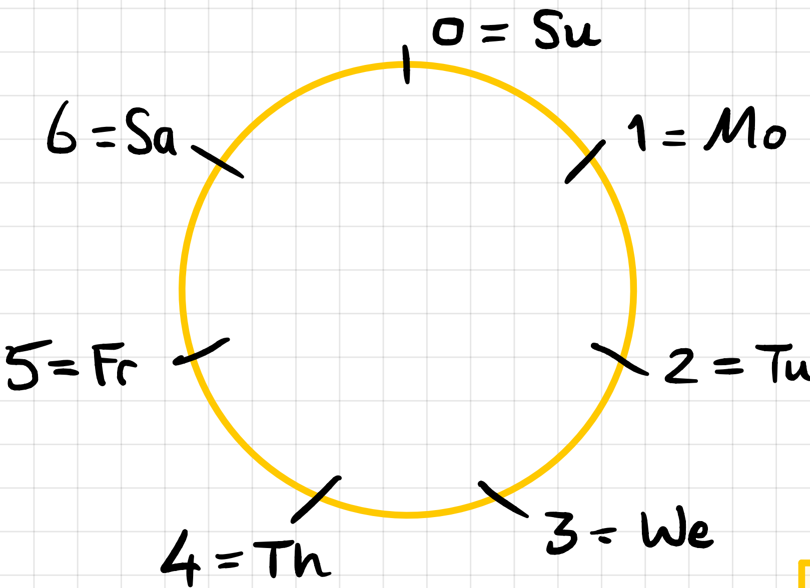
Next topic: Modular Arithmetic

- Modular arithmetic basics
- Computation mod n: addition/multiplication, exponentiation
- GCD and inverses
- Chinese Remainder Theorem
- Fermat's Little Theorem
- Application: RSA Cryptosystem

(More applications later)

Modular arithmetic: arithmetic limited to **integers** in a bounded range $\{0, 1, 2, \dots, m-1\}$ for some m

E.g.: Days of the week $(m = 7)$



Today = Tu Feb. 10

$\boxed{\text{Q:}}$ What day is Apr. 6?

$\boxed{\text{A:}}$

Residue classes (= equivalent numbers) mod 7

$0: \quad \{\ldots, -14, -7, 0, 7, 14, 21, \ldots\}$

$1: \quad \{\ldots, -13, -6, 1, 8, 15, 22, \ldots\}$

$2: \quad \{\ldots, -12, -5, 2, 9, 16, 23, \ldots\}$

$\vdots \qquad\qquad \vdots$

$6: \quad \{\ldots, -8, -1, 6, 13, 20, 27, \ldots\}$

We write (e.g.)

$$23 \equiv 2 \pmod 7 \qquad -1 \equiv 6 \pmod 7$$

$$-14 \equiv 0 \pmod 7$$

Generally: $\boxed{a \equiv b \pmod m \quad \text{iff} \quad m \mid (a-b)}$

Computation mod m

Example:

$$(759 \times 46) + 2268 \pmod 7$$
$$\equiv (3 \times 4) + 0 \pmod 7$$
$$\equiv 12 \pmod 7$$
$$\equiv \boxed{5 \pmod 7}$$

Useful fact: Can always reduce intermediate results mod m $\longrightarrow$ no large numbers!

Formally:

$$(a \pm b) \equiv \big((a \bmod m) \pm (b \bmod m)\big) \bmod m$$
$$ab \equiv \big((a \bmod m) \times (b \bmod m)\big) \bmod m$$

Proof: Ex. or see notes

Exponentiation

Addition & multiplication are efficient (at most $O(n^2)$ where n = # of bits in numbers)

What about exponentiation?

Example: Compute $3^{42} \pmod{53}$

Method 1: $3^2 = 9$; $3^3 = 27$; $3^4 = 81 \equiv 28$

$3^5 = 3 \times 28 = 84 \equiv 31$, ...

# of multiplications $= 41$

generally, $m-1$ multiplicn's to compute x^m

Method 2: [repeated squaring]

Note: # of digits in decimal number x is $\lfloor \log_{10} x \rfloor + 1$

# of bits in binary number x is $\lfloor \log_2 x \rfloor + 1$

Exponentiation

Example: Compute $3^{42} \pmod{53}$

Method 2: [repeated squaring]

$$42 = \overset{32}{1}\,\overset{16}{0}\,\overset{8}{1}\,\overset{4}{0}\,\overset{2}{1}\,\overset{1}{0} \text{ in binary}$$

So $\boxed{3^{42} = 3^{32} \times 3^{8} \times 3^{2}}$

Compute:

$$\left.\begin{aligned}
3^2 &= 9 \\
3^4 &= 9^2 = 81 \equiv 28 \\
3^8 &= 28^2 = 784 \equiv 42 \\
3^{16} &= 42^2 = 1764 \equiv 15 \\
3^{32} &= 15^2 = 225 \equiv 13
\end{aligned}\right\}$$

$$\begin{aligned}
3^{42} &= 3^{32} \times 3^{8} \times 3^{2} \\
&= 13 \times 42 \times 9 \\
&\equiv 13 \times (-11) \times 9 \\
&\equiv 13 \times (-99) \\
&\equiv 13 \times 7 = 91 \equiv \boxed{38}
\end{aligned}$$

\# of multiplications $= 7$!

Generally:

\# of multiplications to compute x^m

$\leq 2 \times$ \# of bits in m

$= O(\log_2 m)$

What about division?

Q: What does $4 \div 3 \pmod 7$ mean?

A: $4 \div 3 \equiv 4 \times (3^{-1}) \pmod 7$, where 3^{-1} is the inverse of $3 \pmod 7$

Defn: x^{-1} (the inverse of x) mod m is the unique number $(\text{mod } m)$ s.t.

$$\boxed{x(x^{-1}) \equiv 1 \pmod m}$$

ALERT: We are assuming here that x^{-1} exists and is unique! Need to prove this!

E.g.: $3 \times 5 = 15 \equiv 1 \pmod 7$

So 5 is the inverse of $3 \pmod 7$

Example: Inverse of 12 mod 35 ?

Example: Inverse of 10 mod 35 ?

$$\left.\begin{aligned}
1 \times 10 &= 10 \\
2 \times 10 &= 20 \\
3 \times 10 &= 30 \\
4 \times 10 &= 40 \equiv 5 \\
5 \times 10 &= 50 \equiv 15 \\
6 \times 10 &= 60 \equiv 25 \\
7 \times 10 &= 35 \equiv 0 \\
8 \times 10 &= 80 \equiv 10 \\
9 \times 10 &= 90 \equiv 30
\end{aligned}\right\} \text{10 has no inverse (mod 35)}$$

ALERT! Two non-zero numbers multiply to 0!

Theorem : $\boxed{x \text{ has an inverse mod } m \iff \gcd(x,m)=1}$

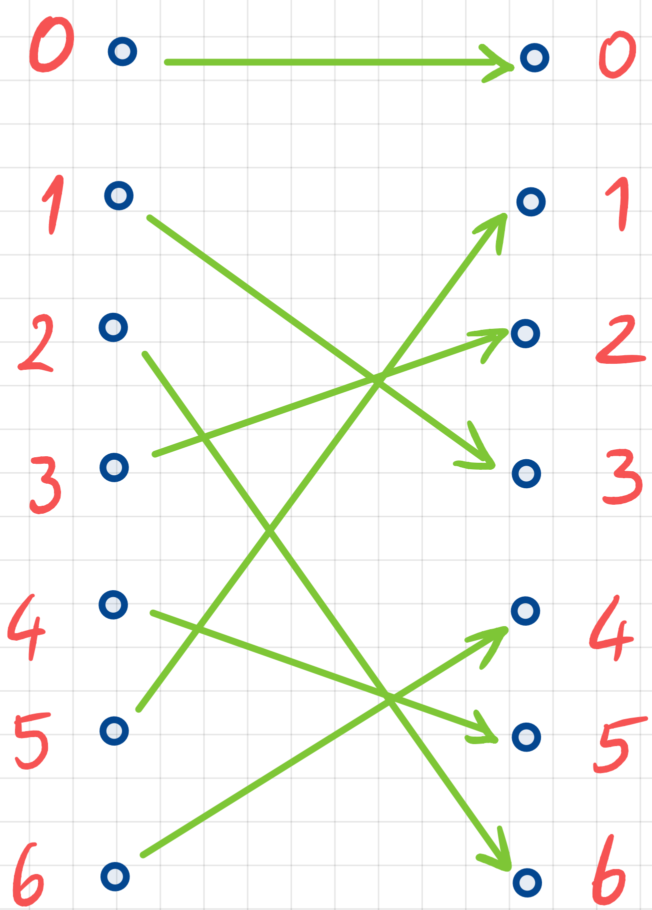
Proof : $\Rightarrow$

Theorem: $\boxed{x \text{ has an inverse mod } m \iff \gcd(x,m)=1}$

Proof: $\Leftarrow$

Example: $m=7$, $x=3$

Picture of the sequence $0, x, 2x, \ldots, 6x$



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

Corollary: Every $x \neq 0$ has an inverse $(\text{mod } p)$ when p is a **prime**

Example: Inverses mod 11

$1^{-1} = 1$

$2^{-1} = 6 \qquad 6^{-1} = 2$

$3^{-1} = 4 \qquad 4^{-1} = 3$

$5^{-1} = 9 \qquad 9^{-1} = 5$

$7^{-1} = 8 \qquad 8^{-1} = 7$

$10^{-1} = 10 \qquad [\equiv (-1) \text{ mod } 11]$

Inverses mod 10

$1^{-1} = 1$

$3^{-1} = 7 \qquad 7^{-1} = 3$

$9^{-1} = 9$

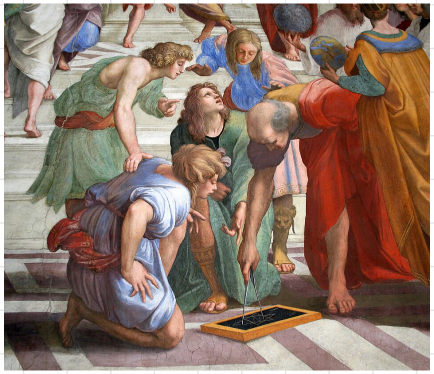
Integers coprime with 10: $\{1, 3, 7, 9\}$

Application of inverses

Solving linear equations mod m

E.g.: $12x \equiv 11 \pmod{35}$

How to compute gcd ?



Euclid's Algorithm

Claim: If $x \geq y > 0$ then

$$\gcd(x,y) = \gcd(y, x \bmod y)$$

Proof: Suffices to prove that

$$d \mid x \ \& \ d \mid y \iff d \mid y \ \& \ d \mid x \bmod y$$

$$x = (x \operatorname{div} y)\, y + (x \bmod y)$$

⊛ follows immediately from this $\square$

Euclid's Algorithm

function gcd(x, y)
  if y = 0 then output(x)
  else output (gcd(y, x mod y))

{assume $x \geq y \geq 0$ & $x > 0$ }

Correctness: strong induction on y (using previous Claim)

Examples:

Euclid's Algorithm

function gcd(x, y)
    if y = 0 then output(x)
    else output (gcd(y, x mod y))

{assume $x \geq y \geq 0$ & $x > 0$ }

Claim: Running time is $O(n)$ where n = # bits in x

Proof:

Computing inverses

Given positive integers x, y, suppose we could compute integers a, b such that

$$\boxed{\gcd(x,y) = ax + by}$$

NOT a modular equation!

Then if $\gcd(x,y) = 1$ we get

$$ax + by = 1$$

and hence

$$by \equiv 1 \pmod{x}$$

So $\boxed{b = y^{-1} \pmod{x}}$ [and $a = x^{-1} \pmod{y}$]

E.g.: $1 = (-1 \times 35) + (3 \times 12)$

$$\Rightarrow \boxed{12^{-1} = 3 \pmod{35}}$$

Extended Euclidean Algorithm

function e_gcd(x,y)
  if y=0 then return(x,1,0)
  else
    (d,a,b) = e_gcd(y, x mod y)
    return (d, A, B) ← $\boxed{d = \gcd(x,y) = Ax + By}$

What should A & B be?

Know: $d = ay + b(x \bmod y)$ (1)

Want: $d = Ax + By$ (2)

From (1): $d = ay + b(x - y(x \operatorname{div} y))$

$= \underbrace{b}_{A}x + \underbrace{(a - (x \operatorname{div} y)b)}_{B}y$

So set $\boxed{A = b}$, $\boxed{B = a - (x \operatorname{div} y)b}$

```
function e_gcd(x,y)
   if y=0 then return(x,1,0)
   else
     (d,a,b) = e_gcd(y, x mod y)
     return (d, b, a-(x div y) b)
```

Example:

e_gcd(35,12)
↓
e_gcd(12,11)
↓
e_gcd(11,1)
↓
e_gcd(1,0)

Correctness & running time analysis as for basic gcd alg.

```
function e_gcd(x,y)
  if y=0 then return(x,1,0)
  else
    (d,a,b) = e_gcd(y, x mod y)
    return (d, b, a-(x div y) b)
```

Example:

e_gcd(35,10) $\downarrow$	(5, 1, −3)	$5 = 1 \cdot 35 + (-3) \cdot 10$
e_gcd(10,5) $\downarrow$	(5, 0, 1) $\uparrow$	$5 = 0 \cdot 10 + 1 \cdot 5$
e_gcd(5,0)	(5, 1, 0) $\uparrow$	$5 = 1 \cdot 5 + 0 \cdot 0$