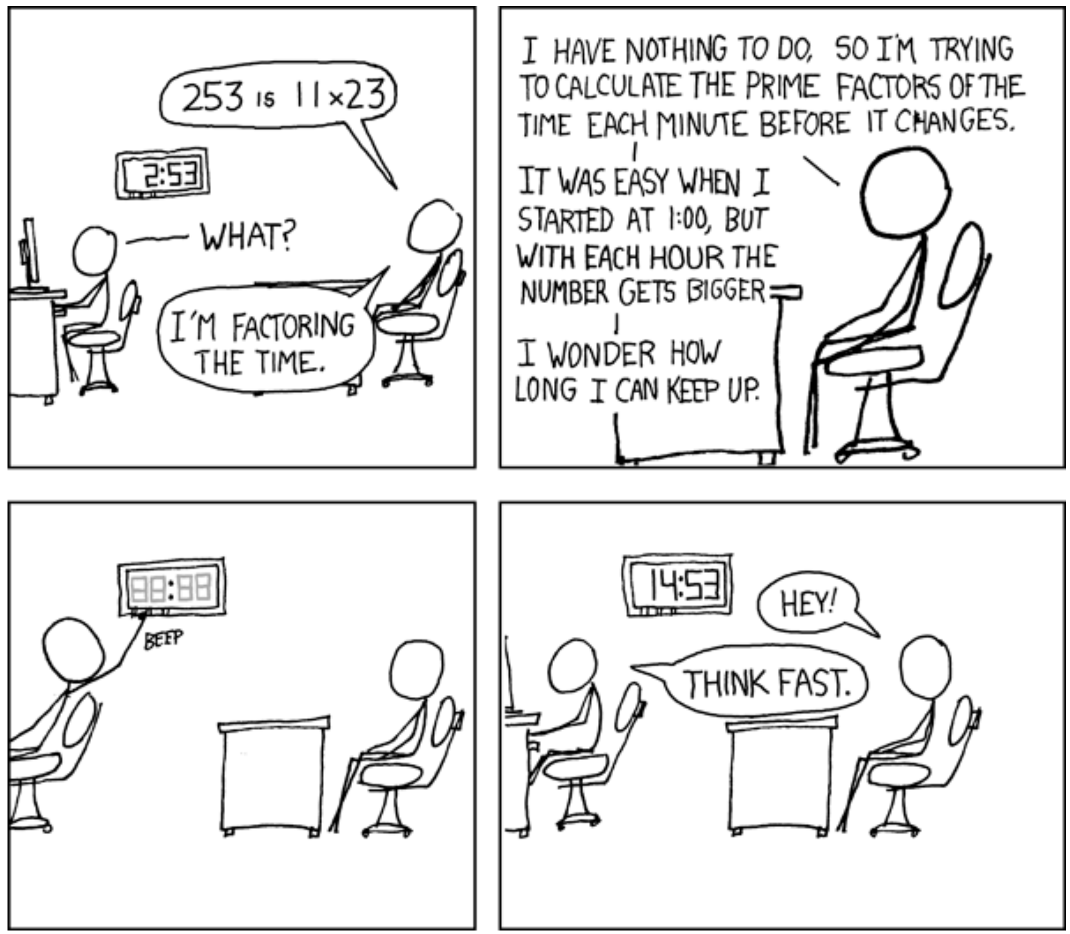
CSE 311: Foundations of Computing

Lecture 12: Primes, GCD



Last Time: Modular Arithmetic

Division Theorem

For $a \in \mathbb{Z}, d \in \mathbb{Z}$ with $d > 0$
there exist *unique* integers *q*, *r* with $0 \leq r < d$
such that $a = dq + r$.

Define "div" by $q = a$ **div** d
and "mod" by $r = a$ **mod** d

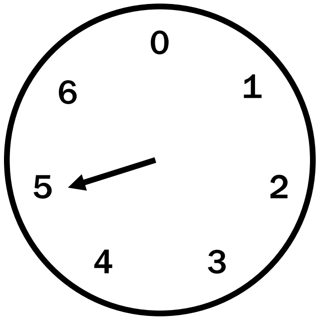
Can then write a **as**

a = (a **div** d) × d + (a **mod** d)

Last Time: Modular Arithmetic

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

$$a \times_7 b = (a \times b) \bmod 7$$



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

Replace number line with a clock. Taking *m* steps returns back to the same place.

Form of arithmetic using only a finite set of numbers {0, 1, 2, 3, …, *m* – 1}

Unclear (so far) that modular arithmetic has the same properties as ordinary arithmetic….

Last Time: Modular Arithmetic

Idea: Find replacement for "=" that works for modular arithmetic

"=" on ordinary numbers allows us to solve problems, e.g.

- add / subtract numbers from both sides of equations
- substitute "=" values in equations

Definition: "a is congruent to b modulo m"

For $a, b, m \in \mathbb{Z}$ with $m > 0$

$$a \equiv b \pmod{m} \leftrightarrow m \mid (a - b)$$

Equivalently, $a \equiv b \pmod{m}$ **iff** $a = b + km$ **for some** $k \in \mathbb{Z}$**.**

Last Time: Modular Arithmetic

Definition: "a is congruent to b modulo m"

For $a, b, m \in \mathbb{Z}$ with $m > 0$

$$a \equiv b \pmod{m} \leftrightarrow m \mid (a - b)$$

Equivalently, $a \equiv b \pmod{m}$ **iff** $a = b + km$ **for some** $k \in \mathbb{Z}$**.**

$\boldsymbol{a \equiv b \pmod{m}}$ if and only if $\boldsymbol{a \bmod m = b \bmod m}$.

I.e., $\boldsymbol{a}$ and $\boldsymbol{b}$ are congruent modulo m iff $\boldsymbol{a}$ and $\boldsymbol{b}$ steps go to the same spot on the "clock" with m numbers

Last Time: Modular Arithmetic: Properties

If $\boldsymbol{a} \equiv \boldsymbol{b} \pmod{\boldsymbol{m}}$ and $\boldsymbol{b} \equiv \boldsymbol{c} \pmod{\boldsymbol{m}}$,
then $\boldsymbol{a} \equiv \boldsymbol{c} \pmod{\boldsymbol{m}}$

If $\boldsymbol{a} \equiv \boldsymbol{b} \pmod{\boldsymbol{m}}$ and $\boldsymbol{c} \equiv \boldsymbol{d} \pmod{\boldsymbol{m}}$,
then $\boldsymbol{a} + \boldsymbol{c} \equiv \boldsymbol{b} + \boldsymbol{d} \pmod{\boldsymbol{m}}$

Corollary: If $\boldsymbol{a} \equiv \boldsymbol{b} \pmod{\boldsymbol{m}}$, then $\boldsymbol{a} + \boldsymbol{c} \equiv \boldsymbol{b} + \boldsymbol{c} \pmod{\boldsymbol{m}}$

If $\boldsymbol{a} \equiv \boldsymbol{b} \pmod{\boldsymbol{m}}$ and $\boldsymbol{c} \equiv \boldsymbol{d} \pmod{\boldsymbol{m}}$,
then $\boldsymbol{ac} \equiv \boldsymbol{bd} \pmod{\boldsymbol{m}}$

Corollary: If $\boldsymbol{a} \equiv \boldsymbol{b} \pmod{\boldsymbol{m}}$, then $\boldsymbol{ac} \equiv \boldsymbol{bc} \pmod{\boldsymbol{m}}$

Last Time: Modular Arithmetic: Properties

If $\boldsymbol{a} \equiv \boldsymbol{b}\ (\mathbf{mod}\ \boldsymbol{m})$ and $\boldsymbol{b} \equiv \boldsymbol{c}\ (\mathbf{mod}\ \boldsymbol{m})$, then $\boldsymbol{a} \equiv \boldsymbol{c}\ (\mathbf{mod}\ \boldsymbol{m})$

If $\boldsymbol{a} \equiv \boldsymbol{b}\ (\mathbf{mod}\ \boldsymbol{m})$, then $\boldsymbol{a} + \boldsymbol{c} \equiv \boldsymbol{b} + \boldsymbol{c}\ (\mathbf{mod}\ \boldsymbol{m})$

If $\boldsymbol{a} \equiv \boldsymbol{b}\ (\mathbf{mod}\ \boldsymbol{m})$, then $\boldsymbol{ac} \equiv \boldsymbol{bc}\ (\mathbf{mod}\ \boldsymbol{m})$

"≡" allows us to solve problems in modular arithmetic, e.g.

- add / subtract numbers from both sides of equations
- chains of "≡" values shows first and last are "≡"
- substitute "≡ " values in equations (not proven yet)

Last Time: Two's Complement

Suppose that $0 \le x < 2^{n-1}$,
x is represented by the binary representation of x
Suppose that $0 \le x \le 2^{n-1}$,
$-x$ is represented by the binary representation of $2^n - x$

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111

Two's complement

Key property: Twos complement representation of any number $\boldsymbol{y}$ is equivalent to $\boldsymbol{y} \bmod \mathbf{2^n}$ so arithmetic works $\bmod \mathbf{2^n}$

Basic Applications of mod

- Two's Complement (last time)
- Hashing
- Pseudo random number generation

Hashing

Scenario:

Map a small number of data values from a large domain $\{0, 1, \ldots, M-1\}$ …

…into a small set of locations $\{0,1, \ldots, n-1\}$ so one can quickly check if some value is present

- $\text{hash}(x) = x \bmod p$ for p a prime close to n
  - or $\text{hash}(x) = (ax + b) \bmod p$
- Depends on all of the bits of the data
  - helps avoid collisions due to similar values
  - need to manage them if they occur

Pseudo-Random Number Generation

Linear Congruential method

$$x_{n+1} = (a\, x_n + c) \bmod m$$

Choose random x_0, a, c, m and produce a long sequence of x_n's

Primality

An integer *p* greater than 1 is called *prime* if the only positive factors of *p* are 1 and *p*.

A positive integer that is greater than 1 and is not prime is called *composite*.

Fundamental Theorem of Arithmetic

Every positive integer greater than 1 has a unique prime factorization

48 = 2 • 2 • 2 • 2 • 3
591 = 3 • 197
45,523 = 45,523
321,950 = 2 • 5 • 5 • 47 • 137
1,234,567,890 = 2 • 3 • 3 • 5 • 3,607 • 3,803

Euclid's Theorem

There are an infinite number of primes.

Proof by contradiction:

Suppose that there are only a finite number of primes and call the full list $p_1, p_2, \ldots, p_n$.

Euclid's Theorem

There are an infinite number of primes.

Proof by contradiction:

Suppose that there are only a finite number of primes and call the full list $p_1, p_2, \ldots, p_n$.

Define the number $P = p_1 \cdot p_2 \cdot p_3 \cdot \cdots \cdot p_n$ and let $Q = P + 1$.

Euclid's Theorem

There are an infinite number of primes.

Proof by contradiction:

Suppose that there are only a finite number of primes and call the full list $p_1, p_2, \ldots, p_n$.

Define the number $P = p_1 \cdot p_2 \cdot p_3 \cdot \cdots \cdot p_n$ and let $Q = P + 1$.

Case 1: Q is prime: Then Q is a prime different from all of $p_1, p_2, \ldots, p_n$ since it is bigger than all of them.

Case 2: $Q > 1$ is not prime: Then Q has some prime factor p (which must be in the list). Therefore $p | P$ and $p | Q$ so $p | (Q - P)$ which means that $p | 1$.

Both cases are contradictions so the assumption is false. ∎

Famous Algorithmic Problems

- Primality Testing
  - Given an integer n, determine if n is prime
- Factoring
  - Given an integer n, determine the prime factorization of n

Factoring

Factor the following 232 digit number [RSA768]:

1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

=

334780716989568987860441698482126908177047949837137685689124313889828837938780022876147116525317430877378144679994‍89

×

367460436667995904282446337996279526322791581643430876426760322838157396665112792333734171433968102700927987363089‍17

Greatest Common Divisor

GCD(a, b):

Largest integer d such that $d \mid a$ and $d \mid b$

- GCD(100, 125) =
- GCD(17, 49) =
- GCD(11, 66) =
- GCD(13, 0) =
- GCD(180, 252) =

GCD and Factoring

$a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46{,}200$

$b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204{,}750$

$$GCD(a, b) = 2^{\min(3,1)} \cdot 3^{\min(1,2)} \cdot 5^{\min(2,3)} \cdot 7^{\min(1,1)} \cdot 11^{\min(1,0)} \cdot 13^{\min(0,1)}$$

Factoring is expensive!

Can we compute GCD(a,b) without factoring?

Useful GCD Fact

If a and b are positive integers, then

$$\gcd(a,b) = \gcd(b, a \bmod b)$$

Useful GCD Fact

If *a* and *b* are positive integers, then
gcd(*a*,*b*) = gcd(*b*, *a* mod *b*)

Proof:

By definition of mod, $a = qb + (a \bmod b)$ for some integer $q = a \operatorname{div} b$.

Let $d = \gcd(a, b)$. Then $d|a$ and $d|b$ so $a = kd$ and $b = jd$ for some integers k and j.

Therefore $(a \bmod b) = a - qb = kd - qjd = (k - qj)d$.
So, $d|(a \bmod b)$ and since $d|b$ we must have $d \leq \gcd(b, a \bmod b)$.

Now, let $e = \gcd(b, a \bmod b)$. Then $e|b$ and $e\,|(a \bmod b)$ so $b = me$ and $(a \bmod b) = ne$ for some integers m and n.

Therefore $a = qb + (a\, mod\, b) = qme + ne = (qm + n)e$.
So, $e|a$ and since $e|b$ we must have $e \leq \gcd(a, b)$.

It follows that $\gcd(a, b) = \gcd(b, a \bmod b)$. ∎

Another simple GCD fact

If a is a positive integer, gcd(a,0) = a.

Euclid's Algorithm

gcd(a, b) = gcd(b, a mod b), gcd(a,0)=a

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

Example: GCD(660, 126)

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) =

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) = gcd(126, 660 mod 126) = gcd(126, 30)
= gcd(30, 126 mod 30) = gcd(30, 6)
= gcd(6, 30 mod 6) = gcd(6, 0)
= 6

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) = gcd(126, 660 mod 126) = gcd(126, 30)
= gcd(30, 126 mod 30) = gcd(30, 6)
= gcd(6, 30 mod 6) = gcd(6, 0)
= 6

In tableau form:

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

Bézout's theorem

If a and b are positive integers, then there exist integers $\boldsymbol{s}$ and $\boldsymbol{t}$ such that

$$\gcd(a,b) = \boldsymbol{s}a + \boldsymbol{t}b.$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

a b b a mod b = r b r

$$\gcd(35, 27) = \gcd(27, 35 \bmod 27) = \gcd(27, 8)$$

a = q * b + r

$$35 = 1 * 27 + 8$$

Extended Euclidean algorithm

- **Can use Euclid's Algorithm to find s, t such that**

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

$$\begin{aligned}
&\overset{a\ \ b}{\gcd(35, 27)} = \overset{b \quad a \bmod b = r}{\gcd(27, 35 \bmod 27)} = \overset{b \quad r}{\gcd(27, 8)} \\
&= \gcd(8, 27 \bmod 8) = \gcd(8, 3) \\
&= \gcd(3, 8 \bmod 3) = \gcd(3, 2) \\
&= \gcd(2, 3 \bmod 2) = \gcd(2, 1) \\
&= \gcd(1, 2 \bmod 1) = \gcd(1, 0)
\end{aligned}$$

a = q * b + r

$$35 = 1 * 27 + 8$$
$$27 = 3 * 8 + 3$$
$$8 = 2 * 3 + 2$$
$$3 = 1 * 2 + 1$$

Extended Euclidean algorithm

- **Can use Euclid's Algorithm to find s, t such that**

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

a = q * b + r

$35 = 1 * 27 + 8$

$27 = 3 * 8 + 3$

$8 = 2 * 3 + 2$

$3 = 1 * 2 + 1$

$2 = 2 * 1 + 0$

r = a – q * b

$8 = 35 – 1 * 27$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

a = q * b + r	r = a – q * b
$35 = 1 * 27 + 8$	$8 = 35 - 1 * 27$
$27 = 3 * 8 + 3$	$3 = 27 - 3 * 8$
$8 = 2 * 3 + 2$	$2 = 8 - 2 * 3$
$3 = 1 * 2 + 1$	$1 = 3 - 1 * 2$
$2 = 2 * 1 + 0$	

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$8 = 35 - 1 * 27$

$3 = 27 - 3 * 8$

$2 = 8 - 2 * 3$

$1 = 3 - 1 * 2$

Plug in the def of 2

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

Re-arrange into 3's and 8's

Extended Euclidean algorithm

- **Can use Euclid's Algorithm to find s, t such that**

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$
$$3 = 27 - 3 * 8$$
$$2 = 8 - 2 * 3$$
$$1 = 3 - 1 * 2$$

Plug in the def of 2

$$1 = 3 - 1 * (8 - 2 * 3)$$
$$= 3 - 8 + 2 * 3$$
$$= (-1) * 8 + 3 * 3$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$= (-1) * 8 + 3 * (27 - 3 * 8)$$
$$= (-1) * 8 + 3 * 27 + (-9) * 8$$
$$= 3 * 27 + (-10) * 8$$

Re-arrange into 8's and 27's

$$= 3 * 27 + (-10) * (35 - 1 * 27)$$
$$= 3 * 27 + (-10) * 35 + 10 * 27$$
$$= 13 * 27 + (-10) * 35$$

Re-arrange into 27's and 35's

Multiplicative inverse $\bmod m$

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

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

$s \bmod m$ is the multiplicative inverse of a:

$$1 = (sa + tm) \bmod m = sa \bmod m$$

Example

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

Example

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

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$26 = 7 * 3 + 5 \qquad 5 = 26 - 7 * 3$$
$$7 = 5 * 1 + 2 \qquad 2 = 7 - 5 * 1$$
$$5 = 2 * 2 + 1 \qquad 1 = 5 - 2 * 2$$

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

Multiplicative inverse of 7 mod 26

Now $(-11) \bmod 26 = 15$. **So,** $x = 15 + 26k$ **for** $k \in \mathbb{Z}$.

Example of a more general equation

Now solve: $7y \equiv 3 \pmod{26}$

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

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

By the multiplicative property of mod we have

$$7 \cdot 15 \cdot 3 \equiv 3 \pmod{26}$$

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

That is, $y = 19 + 26k$ for any integer k is a solution.

Math mod a prime is especially nice

$\gcd(a, m) = 1$ **if** m **is prime and** $0 < a < m$ **so can always solve these equations mod a prime.**

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

X	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

mod 7