Lecture 6: Modular Arithmetic 1

Because Sometimes You Just Want $2 + 2 = 1$

Arithmetic For Days

It is currently Tuesday.
What day is it in 100 days?

7 days from now: Tuesday
14 days from now: Tuesday
21 days from now: Tuesday
...
98 days from now: Tuesday
99 days from now: Wednesday
100 days from now: Thursday!

Phew! There must be a better way...

Week By Week

100 days is 14 weeks and 2 days

Moving 1 week doesn't change day of the week!

So 100 days "equivalent" to 2 days!
2 days from now is Thursday.

What day of the week is it in 2^{100} days?

...
Need more general framework to work with this

Modular Arithmetic

Normally define arithmetic on $\mathbb{Z}$ or $\mathbb{R}$
Now define $+$ and $\cdot$ on $\mathbb{Z}_m := \{0, 1, 2, ..., m-1\}$

Idea: do $+$ or $\cdot$ as normal, shrink down if too big
Ex: for $m = 5$, $3 + 3 = 6 \to 1$; $3 \cdot 4 = 12 \to 2$

What about subtraction?
Really just adding inverses — same idea!
Ex: for $m = 5$, $2 - 4 = 2 + (-4) = -2 \to 3$

What about division?
More complicated...deal with it later

A Quotient View

Say $x \equiv y \pmod{m}$ if $x = y + km$ for $k \in \mathbb{Z}$

Idea: treat such x and y as "the same"
So for $m = 5$, $\{..., -8, -3, 2, 7, ...\}$ all "the same"

$+$ and $\cdot$ now work as normal
Doesn't matter what "representative" used

So for $m = 5$, $42 \cdot 9001$ "same as" $2 \cdot 1 = 1$.

More complicated:

$$(100 + 15) \cdot 6 \equiv (0 + 3) \cdot 2 \equiv 6 \equiv 2 \pmod{4}$$

Well-Defined

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

Proof:

- By givens, $a = c + km$ and $b = d + \ell m$
- So $a + b = c + d + (k + \ell)m$
- Thus $a + b \equiv c + d \pmod{m}$

Can prove similar statement for $\cdot$

Many Days From Now...

Ask now: what day of the week in 2^{100} days?

Need to know $2^{100} \pmod 7$

Notice: $2^3 = 8 \equiv 1 \pmod 7$
So $2^{100} = 2^{99} \cdot 2 = 8^{33} \cdot 2 \equiv 1^{33} \cdot 2 \equiv 2 \pmod 7$

So Thursday again in 2^{100} days!

How to do this in general? Algorithm?

Naïve Approach

Inputs: $x, y, m \in \mathbb{N}$ ($x, m \neq 0$)
Goal Output: $x^y \pmod{m}$

Algorithm:

```
counter, result = 0, 1
while counter ≤ y:
    result = result * x (mod m)
    counter += 1
return result
```

Issue: for applications, y could be 1000+ bits
So could require $\approx 2^{1000}$ iterations
Zzzzz....

Recursive Approach

Idea: If $y = 2k$, $x^y = x^{2k} = (x^k)^2$
If $y = 2k + 1$, $x^y = x^{2k+1} = (x^k)^2 \cdot x$
If can calculate x^k, rest is easy!

Algorithm:

```
mod-exp(x, y, m):
    if y = 0:  return 1
    if y even:
        z = mod-exp(x, y/2, m)
        return z * z (mod m)
    if y odd:
        z = mod-exp(x, (y - 1)/2, m)
        return z * z * x (mod m)
```

Iterative Approach

Alternate approach that may be easier by hand

Idea: decompose y into sum of powers of 2
Ex: 13 is 1101 in binary, so $13 = 2^3 + 2^2 + 2^0$

Note: $(x^{2^i})^2 = x^{2^i \cdot 2} = x^{2^{i+1}}$
So can calculate x raised to powers of two

Algorithm:

- Calculate $x^{2^i} \pmod{m}$ for i up to $\lfloor \log_2(y) \rfloor$
- Multiply those in decomp of y

This is known as the *method of repeated squares*

Repeated Squares Example

Want to calculate $4^{21} \pmod{11}$

$4^1 \equiv 4 \pmod{11}$
$4^2 \equiv 16 \equiv 5 \pmod{11}$
$4^4 \equiv 5^2 \equiv 25 \equiv 3 \pmod{11}$
$4^8 \equiv 3^2 \equiv 9 \pmod{11}$
$4^{16} \equiv 9^2 \equiv 81 \equiv 4 \pmod{11}$

$21 = 16 + 4 + 1$, so $4^{21} = 4^{16} \cdot 4^4 \cdot 4^1$
Thus, $4^{21} \equiv 4 \cdot 3 \cdot 4 \equiv 48 \equiv 4 \pmod{11}$

Move Fast And **Break** Things

Time for a breather! Talk to your neighbors :)

Today's Discussion Question:
If you could have an unlimited storage of one thing, what would it be and why?

Inverses

Return to the problem of division!

In $\mathbb{R}$, $x \div 2$ really just $x \cdot \frac{1}{2}$
What is $\frac{1}{2}$? Number such that $2 \cdot \frac{1}{2} = 1$!

To do division, need *multiplicative inverses*
Mult inverse of x mod m is a st $ax \equiv 1 \pmod{m}$

Claim: If inverse exists, is unique

Proof:

- Suppose have two inverses a and b
- $a \equiv a \cdot 1 \equiv a \cdot (bx) \pmod{m}$
- $b \equiv b \cdot 1 \equiv b \cdot (ax) \pmod{m}$
- Multiplication commutes, so $a \equiv b \pmod{m}$

When Are There Inverses?

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

Proof (only if):

- Proceed by contraposition
- Suppose $\gcd(x, m) = d > 1$
- For any a, $d | ax$ as $d | x$
- For any k, $d | km$ as $d | m$
- Since $d > 1$, $d \not| (km + 1)$
- Hence $ax \neq km + 1$ for any a, k
- So $ax \not\equiv 1 \pmod{m}$ for any a

When Are There Inverses? 2

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

Proof (if):

- Suppose $\gcd(x, m) = 1$
- Consider sequence $0x, 1x, 2x, \ldots, (m-1)x$
- Claim: these are all distinct mod m
  - If $ax \equiv bx \pmod{m}$, $m | ((a-b)x)$
  - $\gcd(x, m) = 1$, so $m | (a-b)$
- m distinct values mod m, so 1 in there!

Calculating GCD

Theorem: For $y > 0$, $\gcd(x, y) = \gcd(y, x \bmod y)$.
Equiv: d divides x and y iff divides y and $x \bmod y$

Proof (only if):

- Suppose $d | x$ and $d | y$, so $x = kd$ and $y = \ell d$
- $x \bmod y = x - qy = d(k - q\ell)$, so $d | (x \bmod y)$

Proof (if):

- Suppose $x \bmod y = jd$ and $y = \ell d$
- $x = (x \bmod y) + qy = d(j + \ell q)$

```
gcd(x, y):
    if y = 0: return x
    else: return gcd(y, x mod y)
```

Example Calculations

$$\begin{aligned}
&\text{Want } \gcd(126, 70) \\
&= \gcd(70, 126 \bmod 70 = 56) \\
&= \gcd(56, 70 \bmod 56 = 14) \\
&= \gcd(14, 56 \bmod 14 = 0) \\
&= 14
\end{aligned}$$

$$\begin{aligned}
&\text{Want } \gcd(70, 61) \\
&= \gcd(61, 70 \bmod 61 = 9) \\
&= \gcd(9, 61 \bmod 9 = 7) \\
&= \gcd(7, 9 \bmod 7 = 2) \\
&= \gcd(2, 7 \bmod 2 = 1) \\
&= \gcd(1, 2 \bmod 1 = 0) \\
&= 1
\end{aligned}$$

Finding Inverses

Knowing GCD good, but would like inverses as well
Brute-force search possible, but slow

Suppose have a, b st $ax + by = \gcd(x, y)$
If gcd $= 1$, $a = x^{-1} \pmod{y}$ and $b = y^{-1} \pmod{x}$!

Why? Have $ax \equiv ax + by \equiv 1 \pmod{y}$

How to find?
Idea: suppose have a', b' st $a'y + b'(x \bmod y) = \gcd$
$x \bmod y = x - \lfloor \frac{x}{y} \rfloor y$
Thus, gcd $= a'y + b'(x - \lfloor \frac{x}{y} \rfloor y) = b'x + (a' - \lfloor \frac{x}{y} \rfloor b')y$

Extended Euclid's Algorithm

Leads to natural extension to Euclid's Algorithm:
egcd(x, y) returns (d, a, b) st gcd $= d = ax + by$

```
egcd(x, y):
    if y = 0: return (x, 1, 0)
    else:
        (d, a', b') = egcd(y, x mod y)
        a = b'
        b = a' - (x//y) * b'
        return (d, a, b)
```

EGCD Example Calculation

If $d = a'y + b'(x \bmod y)$, $d = b'x + (a' - \lfloor \frac{x}{y} \rfloor b')y$

$$
\begin{array}{ll}
\text{egcd}(127,\ 70) & (1, -27, 22 - (\lfloor \frac{127}{70} \rfloor \cdot -27) = 49) \\
\text{egcd}(70,\ 57) & (1, 22, -5 - (\lfloor \frac{70}{57} \rfloor \cdot 22) = -27) \\
\text{egcd}(57,\ 13) & (1, -5, 2 - (\lfloor \frac{57}{13} \rfloor \cdot -5) = 22) \\
\text{egcd}(13,\ 5) & (1, 2, -1 - (\lfloor \frac{13}{5} \rfloor \cdot 2) = -5) \\
\text{egcd}(5,\ 3) & (1, -1, 1 - (\lfloor \frac{5}{3} \rfloor \cdot -1) = 2) \\
\text{egcd}(3,\ 2) & (1, 1, 0 - (\lfloor \frac{3}{2} \rfloor \cdot 1) = -1) \\
\text{egcd}(2,\ 1) & (1, 0, 1 - (\lfloor \frac{2}{1} \rfloor \cdot 0) = 1) \\
\text{egcd}(1,\ 0) & (1, 1, 0)
\end{array}
$$

So $\text{gcd}(127, 70) = 1 = (-27 \cdot 127) + (49 \cdot 70)$

Fin

Next time: yet more modular arithmetic!