Computing inverses.

How to find the inverse?

How to find if $x$ has an inverse modulo $m$?

Find gcd $(x, m)$.

Greater than 1? No multiplicative inverse.

Equal to 1? Multiplicative inverse.

Algorithm: Try all numbers up to $x$ to see if it divides both $x$ and $m$.

Very slow.

Greatest Common Divisor and Inverses.

Thm:
If greatest common divisor of $x$ and $m$, gcd$(x, m)$, is 1, then $x$ has a multiplicative inverse modulo $m$.

Proof
Claim: The set $S = \{0.x.1.x.\ldots.(m-1)x\}$ contains $y = 1 \mod m$ if all distinct modulo $m$.

Each of $m$ numbers in $S$ correspond to one of $m$ equivalence classes modulo $m$.

$\implies$ One must correspond to 1 modulo $m$. Inverse Exists!

Proof of Claim: If not distinct, then $\exists a,b \in \{0,\ldots,m-1\}, a \neq b$, where $(ax = bx \mod (m)) \implies (a-b)x = 0 \mod (m)$

Or $(a-b)x = km$ for some integer $k$.

$\gcd(x, m) = 1$

$\implies$ Prime factorization of $m$ and $x$ do not contain common primes.

$\implies (a-b)$ factorization contains all primes in $m$'s factorization.

So $(a-b)$ has to be multiple of $m$.

$\implies (a-b) \geq m$. But $a,b \in \{0,\ldots,m-1\}$. Contradiction.

Only if

Thm: If gcd$(x, m) \neq 1$ then $x$ has no multiplicative inverse modulo $m$.

Assume there is inverse $a$ is $x^{-1}$, or $ax = 1 + km$.

Thus,

$a(nd) = 1 + kd$ or

$d(na - kℓ) = 1$.

But $d > 1$ and $z = (na - kℓ) \notin \mathbb{Z}$.

so $dz \neq 1$ and $dz = 1$. Contradiction.

Euclid’s algorithm.

GCD Mod Corollary: gcd$(x, y) = \gcd(y, \mod (x, y))$.

Hey, what’s gcd$(7, 0)$? 7 since 7 divides 7 and 7 divides 0

What’s gcd$(x, 0)$?

\[\begin{align*}
\text{gcd}(x, y) = \gcd(y, \mod (x, y))
\end{align*}\]

Theorem: (euclid x y) = gcd$(x, y)$ if $x \geq y$.

Proof: Use Strong Induction.

Base Case: $y = 0$, “$x$ divides $y$ and $x$”

\[\implies x \text{ is common divisor and clearly largest.}\]

Induction Step: $\mod (x, y) < y < x$ when $x \geq y$

call in line (***) meets conditions plus arguments “smaller”

and by strong induction hypothesis computes gcd$(y, \mod (x, y))$ which is gcd$(x, y)$ by GCD Mod Corollary.
Excursion: Value and Size.

Before discussing running time of gcd procedure...
What is the value of 1,000,000?
one million or 1,000,000!  
What is the “size” of 1,000,000?  
Number of digits in base 10: 7.  
Number of bits (a digit in base 2): 21.  
For a number \(x\), what is its size in bits?  
\[ n = b(x) \approx \log_2 x \]

Euclid procedure is fast.

**Theorem:** \((\text{euclid } x \ y)\) uses \(2^n\) “divisions” where \(n = b(x) \approx \log_2 x\).

Is this good? Better than trying all numbers in \([2 \ldots y/2]\)?  
Check 2, check 3, check 4, check 5 . . . , check \(y/2\).  
If \(y \approx x\) roughly \(y\) uses \(n\) bits . . .  
\(2^{n-1}\) divisions! Exponential dependence on size!  
101 bit number. \(2^{100} \approx 10^{30} = \) “million, trillion, trillion” divisions!  
\(2n\) is much faster! . . . roughly 200 divisions.

Poll.

Assume \(\log_2 1,000,000\) is 20 to the nearest integer.  
Mark what’s true.  
(A) The size of 1,000,000 is 20 bits.  
(B) The size of 1,000,000 is one million.  
(C) The value of 1,000,000 is one million.  
(D) The value of 1,000,000 is 20.  
(A) and (C).

Poll

-Algorithms at work.

Which are correct?
(A) \(\text{gcd}(700,568) = \text{gcd}(568,132)\)  
(B) \(\text{gcd}(8,3) = \text{gcd}(3,2)\)  
(C) \(\text{gcd}(8,3) = 1\)  
(D) \(\text{gcd}(4,0) = 4\)

Try everything
Check 2, check 3, check 4, check 5 . . . , check \(y/2\).
“(gcd x y)” at work.
\[
\text{euclid}(700,568) \\
\text{euclid}(568, 132) \\
\text{euclid}(132, 40) \\
\text{euclid}(40, 12) \\
\text{euclid}(12, 4) \\
\text{euclid}(4, 0) \\
4
\]
Notice: The first argument decreases rapidly.  
At least a factor of 2 in two recursive calls.  
(The second is less than the first.)

Runtime Proof.

\[
\text{(define (euclid } x \ y) (if (= y 0) x (euclid y (mod x y))))
\]

**Theorem:** \((\text{euclid } x \ y)\) uses \(O(n)\) “divisions” where \(n = b(x)\).

**Proof:**

**Fact:**
First arg decreases by at least factor of two in two recursive calls.  
After \(2\log_2 x = O(n)\) recursive calls, argument \(x\) is 1 bit number.  
One more recursive call to finish.  
1 division per recursive call.  
\(O(n)\) divisions.

12/35
Runtime Proof (continued.)

(define (euclid x y)
  (if (= y 0)
      x
      (euclid y (mod x y)))))

Fact:
First arg decreases by at least factor of two in two recursive calls.

Proof of Fact:
Recall that first argument decreases every call.
Case 1: \( y < \frac{x}{2} \), first argument is \( y \) => true in one recursive call;
Case 2: Will show \( y \geq \frac{x}{2} \) => \( \text{mod}(x,y) \leq \frac{x}{2} \).

\( \text{mod} (x,y) \) is second argument in next recursive call, and becomes the first argument in the next one.
When \( y \geq \frac{x}{2} \), then \( \lfloor \frac{x}{y} \rfloor = 1 \),
\( \text{mod} (x,y) = x - y \lfloor \frac{x}{y} \rfloor = x - y \leq x - \frac{x}{2} = \frac{x}{2} \)

Euclid’s GCD algorithm.

(define (euclid x y)
  (if (= y 0)
      x
      (euclid y (mod x y)))))
Computes the gcd\((x,y)\) in \( O(n) \) divisions. (Remember \( n = \log_2 x \).)

For \( x \) and \( m \), if gcd\((x,m)\) = 1 then \( x \) has an inverse modulo \( m \).

Multiplicative Inverse.

GCD algorithm used to tell if there is a multiplicative inverse.
How do we find a multiplicative inverse?

Finding an inverse?

We showed how to efficiently tell if there is an inverse.
Extend euclid to find inverse.

Extended GCD

Euclid’s Extended GCD Theorem: For any \( x,y \) there are integers \( a,b \) such that
\[ ax + by = d \]
where \( d = \text{gcd}(x,y) \).

“Make \( d \) out of sum of multiples of \( x \) and \( y \).”

What is multiplicative inverse of \( x \) modulo \( m \)?
By extended GCD theorem, when \( \text{gcd}(x,m) = 1 \),
\[ ax + bm = 1 \]
\[ ax \equiv 1 \pmod{m} \]
So \( a \) multiplicative inverse of \( x \) \( \pmod{m} \)!!
Example: For \( x = 12 \) and \( y = 35 \), \( \text{gcd}(12,35) = 1 \).
\[ (3)12 + (−1)35 = 1 \]
\[ a = 3 \text{ and } b = −1. \]
The multiplicative inverse of 12 \( \pmod{35} \) is 3.
Check: \( 3(12) \equiv 36 \equiv 1 \pmod{35} \).

Poll

Mark correct answers.
Note: \( \text{Mod}(x,y) \) is the remainder of \( x \) divided by \( y \) and \( y < x \).

(A) \( \text{mod} (x,y) < y \)
(B) If \( \text{euclid}(x,y) \) calls \( \text{euclid}(u,v) \) calls \( \text{euclid} (a,b) \) then \( a < \frac{x}{2} \).
(C) \( \text{euclid}(x,y) \) calls \( \text{euclid} (u,v) \) means \( u = y \).
(D) if \( y > \frac{x}{2} \), \( \text{mod} (x,y) = (x − y) \)
(E) if \( y > \frac{x}{2} \), \( \text{mod} (x,y) < \frac{x}{2} \)

13 / 35

14 / 35

15 / 35

16 / 35

17 / 35

18 / 35

19 / 35
Make $d$ out of multiples of $x$ and $y$.

$$\gcd(35,12)$$
$$\gcd(12, 11) \quad \Rightarrow \quad \gcd(12, 12\%12)$$
$$\gcd(1, 0)$$

How did $\gcd$ get 11 from 35 and 12?

$\gcd(35,12)$
$$\gcd(12, 11) \quad \Rightarrow \quad \gcd(12, 12\%12)$$
$$\gcd(1, 0)$$

How does $\gcd$ get 1 from 12 and 11?

$\gcd(12, 11) \quad \Rightarrow \quad \gcd(12, 35\%12)$
$$\gcd(1, 0)$$

Algorithm finally returns 1.

Get 11 from 35 and 12 and plug in...

$\gcd(35,12)$
$$\gcd(12, 11) \quad \Rightarrow \quad \gcd(12, 35\%12)$$
$$\gcd(1, 0)$$

Get 11 from 35 and 12 and plug in.

Simplify. $a = 3$ and $b = -1$.

**Correctness.**

**Proof:** Strong Induction.\(^1\)

**Base:** $\text{ext-gcd}(x,0)$ returns $(d = x,1,0)$ with $x = (1)x + (0)y$.

**Induction Step:** $\text{ext-gcd}(d,A,B)$ with $d = Ax + By$

Ind hyp: $\text{ext-gcd}(y, \mod(x,y))$ returns $(d,a,b)$ with

$d = ay + bx \mod(x,y)$

$\text{ext-gcd}(x,y)$ calls $\text{ext-gcd}(y, \mod(x,y))$ so

$d = ay + bx \mod(x,y)$

Recursively: $d = ay + bx - (a - \lfloor x/y \rfloor b)y$

Returns $(d,b,(a - \lfloor x/y \rfloor \cdot b))$.

And $\text{ext-gcd}$ returns $(d,b,(a - \lfloor x/y \rfloor \cdot b))$ so theorem holds! \(\square\)

\(^1\)Assume $d = \gcd(x,y)$ by previous proof.

### Extended GCD Algorithm.

$\text{ext-gcd}(x,y)$

if $y = 0$ then return$(x, 1, 0)$
else

$(d,a,b) := \text{ext-gcd}(y, \mod(x,y))$

return $(d, b, a - \lfloor x/y \rfloor \cdot b)$

Claim: Returns $(d,a,b)$; $d = \gcd(a,b)$ and $d = ax + by$.

**Example:**

$\text{gcd}(7,60) = 1$.

$\text{egcd}(7,60)$

$7(0) + 60(1) = 60$

$7(1) + 60(0) = 7$

$7(-8) + 60(1) = 4$

$7(9) + 60(-1) = 3$

$7(-17) + 60(2) = 1$

**Theorem:** Returns $(d,a,b)$, where $d = \gcd(a,b)$ and

$d = ax + by$.

**Review Proof:** step.

$\text{ext-gcd}(x,y)$

if $y = 0$ then return$(x, 1, 0)$
else

$(d,a,b) := \text{ext-gcd}(y, \mod(x,y))$

return $(d, b, a - \lfloor x/y \rfloor \cdot b)$

Returns $(d,b,(a - \lfloor x/y \rfloor \cdot b))$.

Hand Calculation Method for Inverses.

**Example:**

$\gcd(7,60) = 1$.

$\text{egcd}(7,60)$

$7(0) + 60(1) = 60$

$7(1) + 60(0) = 7$

$7(-8) + 60(1) = 4$

$7(9) + 60(-1) = 3$

$7(-17) + 60(2) = 1$

Confirm: $-119 + 120 = 1$

Note: an “iterative” version of the e-gcd algorithm.
Wrap-up

Conclusion: Can find multiplicative inverses in $O(n)$ time!
Very different from elementary school: try 1, try 2, try 3...
$2^{n/2}$

Inverse of 500,000,357 modulo 1,000,000,000,000?
< 80 divisions.
versus 1,000,000

Internet Security.
Public Key Cryptography: 512 digits.
512 divisions vs.
(10000000000000000000000000000000000000000000)² divisions.

Internet Security: Soon.

Simple Chinese Remainder Theorem.

My love is won. Zero and One. Nothing and nothing done.
Find $x = a \pmod{m}$ and $x = b \pmod{n}$ where $\gcd(m,n) = 1$.

**CRT Thm**: There is a unique solution $x \pmod{mn}$.

**Proof (solution exists)**:
Consider $u = n^{n-1} \pmod{m}$.
$u \equiv 0 \pmod{n}$ $u \equiv 1 \pmod{m}$

Consider $v = m^{m-1} \pmod{n}$.
$v \equiv 1 \pmod{m}$ $v \equiv 0 \pmod{n}$

Let $x = au + bv$.
$x = a \pmod{m}$ since $bv \equiv 0 \pmod{m}$ and $au \equiv a \pmod{m}$
$x = b \pmod{n}$ since $au \equiv 0 \pmod{n}$ and $bv \equiv b \pmod{n}$
This shows there is a solution.

Simple Chinese Remainder Theorem.

**CRT Thm**: There is a unique solution $x \pmod{mn}$.

**Proof (uniqueness)**:
If not, two solutions, $x$ and $y$.
$(x - y) \equiv 0 \pmod{m}$ and $(x - y) \equiv 0 \pmod{n}$.

$x - y \text{ is multiple of } m \text{ and } n$

$\gcd(m,n) = 1 \implies \text{no common primes in factorization } m \text{ and } n$

$\implies mn | (x - y)$

$x - y \geq mn \implies x, y \notin [0, \ldots, mn - 1]$.  

Thus, only one solution modulo $mn$.

Bijections

**Bijection is one to one and onto.**

Bijection:
$f: A \to B$.
Domain: $A$, Co-Domain: $B$.
Versus Range.
E.g. $\sin(x)$.
$A - B = \text{ reals}$.
Range is $[-1,1]$. Onto: $[-1,1]$.
Not one-to-one: $\sin(x) = \sin(0) = 0$.

Range Definition always is onto.
Consider $f(x) = ax \pmod{m}$.
$f: \{0, \ldots, m-1\} \to \{0, \ldots, m-1\}$.
Domain/Co-Domain: $\{0, \ldots, m-1\}$.

When is it a bijection?
When $\gcd(a,m)$ is ....? ... 1.

Not Example: $a = 2, m = 4, f(0) = f(2) = 0 \pmod{4}$.

Lots of Mods

$x = 5 \pmod{7}$ and $x = 3 \pmod{5}$.

What is $x \pmod{35}$? Let’s try 5. Not 3 (mod 5)!
Let’s try 3. Not 5 (mod 7)!
If $x = 5 \pmod{7}$ then $x$ is in $\{5,12,19,26,33\}$.

Oh, only 33 is 3 (mod 5).

Hmmm... only one solution.

A bit slow for large values.

Poll.

My love is won, Zero and one. Nothing and nothing done.
What is the rhyme saying?

(A) Multiplying by 1, gives back number. (Does nothing.)
(B) Adding 0 gives back number. (Does nothing.)
(C) Rao has gone mad.
(D) Multiplying by 0, gives 0.
(E) Adding one does, not too much.

All are (maybe) correct.
(E) doesn’t have to do with the rhyme.
(C) Recall Polonius: “Though this be madness, yet there is method in ‘t.”
CRT: isomorphism.

For \( m, n \), gcd\( (m, n) = 1 \).
\[ x \mod mn \leftrightarrow x \mod m \text{ and } x = b \mod n \]
\[ y \mod mn \leftrightarrow y \mod m \text{ and } y = d \mod n \]
Also, true that \( x + y \mod mn \leftrightarrow a + c \mod m \text{ and } b + d \mod n \).
Mapping is "isomorphic": corresponding addition (and multiplication) operations consistent with mapping.

Fermat and Exponent reducing.

**Fermat’s Little Theorem:** For prime \( p \), and \( a \not\equiv 0 \pmod{p} \),
\[ a^{p-1} \equiv 1 \pmod{p} \]

**Proof:** Consider \( S = \{a, 1, \ldots, a (p-1)\} \).
All different modulo \( p \) since \( a \) has an inverse modulo \( p \).
\( S \) contains representative of \( \{1, \ldots, p-1\} \) modulo \( p \).
\[ (a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) \equiv 1 \cdot 2 \cdot (p-1) \pmod{p} \]
Since multiplication is commutative.
\[ a^{p-1}(1 \cdots (p-1)) = (1 \cdot \cdots (p-1)) \pmod{p} \]
Each of \( 2 \cdots (p-1) \) has an inverse modulo \( p \), solve to get...
\[ a^{p-1} \equiv 1 \pmod{p} \]

Poll

Which was used in Fermat’s theorem proof?
(A) The mapping \( f(x) = a x \mod p \) is a bijection.
(B) Multiplying a number by 1, gives the number.
(C) All nonzero numbers \( \pmod{p} \) have an inverse.
(D) Multiplying a number by 0 gives 0.
(E) Multiplying elements of sets \( A \) and \( B \) together is the same if \( A = B \).
(A), (C), and (E)

Lecture in a minute.

Euclid’s Alg: \( \gcd(x, y) = \gcd(y, x \pmod{y}) \)
Fast cuz value drops by a factor of two every two recursive calls.

Extended Euclid: Find \( a, b \) where \( ax + by = \gcd(x, y) \).
\[ \text{Idea: compute } a, b \text{ recursively (euclid), or iteratively.} \]
Inverse: \( ax + by = \gcd(x, y) \pmod{y} \).
If \( \gcd(x, y) = 1 \), we have \( ax = 1 \pmod{y} \)
\[ \Rightarrow a = x^{-1} \pmod{y} \]

Chinese Remainder Theorem:
If \( \gcd(n, m) = 1 \), \( x = a \pmod{n}, x = b \pmod{m} \) unique sol.

Proof: Find \( u = 1 \pmod{m} \), \( u = 0 \pmod{n} \),
and \( v = 0 \pmod{n} \), \( v = 1 \pmod{m} \).
Then: \( x = au + bv = a \pmod{m} \)
\[ u = m(m^{-1} \pmod{n}) \pmod{m} \text{ works!} \]
Fermat: Prime \( p \), \( a^{p-1} \equiv 1 \pmod{p} \).

Proof Idea: \( f(x) = a(x) \pmod{p} \): bijection on \( S = \{1, \ldots, p-1\} \).
Product of elts \( \equiv \pmod{\text{range/domain: } a^{p-1}} \) factor in range.