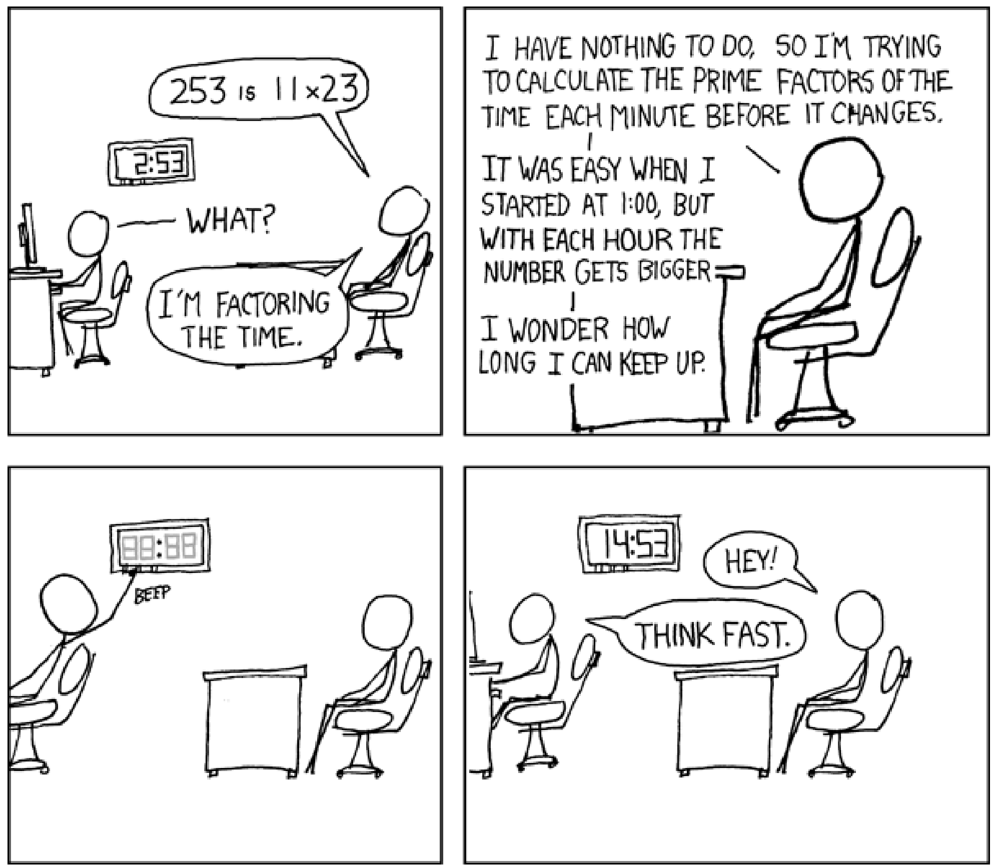
CSE 311: Foundations of Computing

Lecture 12: Two's Complement, Primes, GCD

Check Solutions
& Feedback on HW
especially HW3 - #5
(also class email from
last monday)

I am Paul Beame.
Shayan has mostly
lost his voice.
He will have office hours
today.

Last time: mod function
mod predicate
mod is nice for
addition
multiplication



n-bit Unsigned Integer Representation

- Represent integer x as sum of powers of 2:

  If $\sum_{i=0}^{n-1} b_i 2^i$ where each $b_i \in \{0,1\}$

  then representation is $b_{n-1} \ldots b_2\ b_1\ b_0$

  2^1 2^0

  … 8 4 2 1

99 = 64 + 32 + 2 + 1

18 = 16 + 2

- For n = 8:

  99: 0110 0011

  \+ 18: 0001 0010

  117: 0111 0101

Sign-Magnitude Integer Representation

n-bit signed integers

Suppose that $-2^{n-1} < x < 2^{n-1}$

First bit as the sign, $n-1$ bits for the value

99 = 64 + 32 + 2 + 1

18 = 16 + 2

For n = 8:

99: 0110 0011

\+ -18: 1001 0010

= -117 : 1111 0101

1000 0 -0

0000 0 : 0

both 0 and -0

Any problems with this representation?

Two's Complement Representation

n bit signed integers, first bit will still be the sign bit

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

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

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

99 = 64 + 32 + 2 + 1
18 = 16 + 2

$$\begin{array}{r} 256 \\ -\ 18 \\ \hline 234 \end{array}$$

For n = 8:
99: 0110 0011
-18: 1110 1110

127 + 64 + 32 + 8 + 4 + 2

Sign-Magnitude vs. Two's Complement

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

Sign-bit

-0

1000

-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

Two's Complement Representation

- For $0 < x \leq 2^{n-1}$, $-x$ is represented by the binary representation of $2^n - x$
  - That is, the two's complement representation of any number y has the same value as y modulo 2^n.

$$2^n - 1 = \longrightarrow \begin{array}{r} 11111111 \\ 00010010 \\ \hline 11101101 \end{array}$$

- To compute this: Flip the bits of x then add 1:
  - All 1's string is $2^n - 1$, so

    Flip the bits of x $\equiv$ replace x by $2^n - 1 - x$

    Then add 1 to get $2^n - x$

Basic Applications of mod

- Hashing
- Pseudo random number generation
- Simple cipher

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

- $\mathrm{hash}(x) = x \bmod p$ for p a prime close to n
  - or $\mathrm{hash}(x) = (ax + b) \bmod p$
- 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

Simple Ciphers

- **Caesar cipher**, A = 1, B = 2, . . .
  - HELLO WORLD
- **Shift cipher**
  - $f(p) = (p + k) \bmod 26$
  - $f^{-1}(p) = (p - k) \bmod 26$
- **More general**
  - $f(p) = (ap + b) \bmod 26$

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 $p_1, p_2, \dots, p_n$.

define $P = p_1 \cdot p_2 \cdot p_3 \cdots p_n$

let $Q = P + 1$

$P + p_1 > Q$

$P + p_2 > Q$

$\vdots$

$P + p_n > Q$

None of $p_1 \cdots p_n$ divides Q

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.

by Fundamental Theorem

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]:

123018668453011775513049495838496272077
285356959533479219732245215172640050726
365751874520219978646938995647494277406
384592519255732630345373154826850791702
612214291346167042921431160222124047927
4737794080665351419597459856902143413

12301866845301177551304949583849627207728535695953347
92197322452151726400507263657518745202199786469389956
47494277406384592519255732630345373154826850791702612
21429134616704292143116022212404792747377940806653514
19597459856902143413

$=$

334780716989568987860441698482126908177047949837
137685689124313889828837938780022876147116525317
43087737814467999489

$\times$

3674604366679959042824463379962795263222791581643
430876426760322838157396665112792333734171433968
10270092798736308917

Greatest Common Divisor

GCD(a, b):

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

- GCD(100, 125) = 25
- GCD(17, 49) = 1
- GCD(11, 66) = 11
- GCD(13, 0) = 13
- GCD(180, 252) = 36

$2 \cdot 2 \cdot 3 \cdot 3 \cdot 5$

$3 \cdot 3 \cdot 28$

$3 \cdot 3 \cdot 2 \cdot 2 \cdot 7$

$$\begin{array}{r|l} & 28 \\ \hline 9 & 252 \\ & 18 \\ \hline & 72 \end{array}$$

GCD and Factoring

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

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

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

$= 2 \cdot 3 \cdot 5^2 \cdot 7 \cdot = 150 \cdot 7 \cdot = 1050$

Factoring is expensive!

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

Useful GCD Fact

$a > b$

If *a* and *b* are positive integers, then

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

Proof:

$a = q \cdot b + (a \bmod b)$ for some $q = a \text{ div } b$

Let $e = \gcd(b, a \bmod b)$: $e \mid b$ and $e \mid (a \bmod b)$

multiple of e — multiple of e

$\therefore e \mid a$ $\therefore e \le \gcd(a,b)$

Let $d = \gcd(a,b)$ $\therefore d \mid a$ and $d \mid b$

$a \bmod b = a - q \cdot b$

$\therefore d \mid (a \bmod b)$ $\therefore d \le \gcd(b, a \bmod b)$

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 $a = qb + (a \bmod b) = qme + ne = (qm + n)e$.
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)

```
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) =

a b a mod b

$660 = 5 \cdot 126 + 30$

$126 = 4 \cdot 30 + \boxed{6}$

$30 = 5 \cdot 6 + 0$

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

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 Intermediary Information):

$$\underbrace{\gcd(35, 27)}_{\gcd(a,\ b)} = \underbrace{\gcd(27, 35 \bmod 27)}_{\gcd(b,\ a \bmod b = r)} = \underbrace{\gcd(27, 8)}_{\gcd(b,\ r)} \quad \underbrace{(35 = 1 * 27 + 8)}_{a = q * b + r}$$

Extended Euclidean algorithm

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

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

Step 1 (Compute GCD & Keep Intermediary Information):

a b b a mod b = r b r a = q * b + r

$$\gcd(35, 27) = \gcd(27, 35 \bmod 27) = \gcd(27, 8) \quad (35 = 1 * 27 + 8)$$

$$= \gcd(8, 27 \bmod 8) = \gcd(8, 3) \quad (27 = 3 * 8 + 3)$$

$$= \gcd(3, 8 \bmod 3) = \gcd(3, 2) \quad (8 = 2 * 3 + 2)$$

$$= \gcd(2, 3 \bmod 2) = \gcd(2, 1) \quad (3 = 1 * 2 + 1)$$

$$= \gcd(1, 2 \bmod 1) = \gcd(1, 0) \quad 2 = 2 \cdot 1 + 0$$

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

Extended Euclidean algorithm

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

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

Step 3 (Backward Substitute Equations):

Plug in the def of 2

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

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

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):

$8 = 35 - 1 * 27$

$3 = 27 - 3 * 8$

$2 = 8 - 2 * 3$

$\boxed{1 = 3 - 1 * 2}$

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

Plug in the def of 3

$$\begin{aligned}
&= (-1) * 8 + 3 * (27 - 3 * 8) \\
&= (-1) * 8 + 3 * 27 + (-9) * 8 \\
&= 3 * 27 + (-10) * 8
\end{aligned}$$

Re-arrange into 8's and 27's

$$\begin{aligned}
&= 3 * 27 + (-10) * (35 - 1 * 27) \\
&= 3 * 27 + (-10) * 35 + 10 * 27 \\
&= \underline{13 * 27} + (-10) * \underline{35}
\end{aligned}$$

Re-arrange into 27's and 35's

t

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

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

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

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

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