Lecture 12: Two’s Complement, Primes, GCD

Check Solutions & Feedback on HW especially HW 3 - 4
(also class email from last Monday)

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

Last term's usual final mod predicate
\( \text{mod } i \) was for addition multiplicative
n-bit Unsigned Integer Representation

- Represent integer $x$ as sum of powers of 2:
  
  If $\sum_{i=0}^{n-1} b_i2^i$ where each $b_i \in \{0,1\}$
  
  then representation is $b_{n-1}...b_2 b_1 b_0$

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

- For $n = 8$:
  
  99: 0110 0011
  18: 0001 0010
  
  $+ \quad +$
  
  117: 0101 1010
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 \(n = 8\):
\[99: \quad 0110 \ 0011\]
\[-18: \quad 1001 \ 0010\]

Any problems with this representation?
Two’s Complement Representation

\( n \) bit signed integers, first bit will still be the sign bit

Suppose that \( 0 \leq x < 2^{n-1} \), \( x \) is represented by the binary representation of \( x \).

Suppose that \( 0 < x \leq 2^{n-1} \), \(-x\) is represented by the binary representation of \( 2^n - x \).

Key property: Twos complement representation of any number \( y \) is equivalent to \( y \mod 2^n \) so arithmetic works \( \mod 2^n \)

\[
\begin{align*}
99 &= 64 + 32 + 2 + 1 \\
18 &= 16 + 2 \\
\text{For } n = 8: \\
99 &= 0110 0011 \\
-18 &= 1110 1110
\end{align*}
\]
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

-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 $2^n$.

• 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

- hash(\(x\)) = \(x \mod p\) for \(p\) a prime close to \(n\)
  - or hash(\(x\)) = \((ax + b) \mod p\)
- 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

\[ x_{n+1} = (a \cdot x_n + c) \mod m \]

Choose random \( x_0, a, c, m \) and produce a long sequence of \( x_n \)'s
Simple Ciphers

- **Caesar cipher,** \( A = 1, B = 2, \ldots \)
  - HELLO WORLD

- **Shift cipher**
  - \( f(p) = (p + k) \mod 26 \)
  - \( f^{-1}(p) = (p - k) \mod 26 \)

- **More general**
  - \( f(p) = (ap + b) \mod 26 \)
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

\[48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3\]
\[591 = 3 \cdot 197\]
\[45,523 = 45,523\]
\[321,950 = 2 \cdot 5 \cdot 5 \cdot 47 \cdot 137\]
\[1,234,567,890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3,607 \cdot 3,803\]
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 $p_1, p_2, \ldots, p_n$.

Define $P = p_1 \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_n$

Let $q = P + 1$

$p_1 | p + 1 > q$

$p_2 | p + 1 > q$

$p_3 | p + 1 > q$

$p_n | p + 1 > q$

None of $p_1 \ldots p_n$ divide $q$
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 $P = p_1 \cdot p_2 \cdot p_3 \cdots \cdot p_n$ 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]:

123018668453011775513049495838496272077
285356959533479219732245215172640050726
365751874520219978646938995647494277406
384592519255732630345373154826850791702
612214291346167042921431160222124047927
4737794080665351419597459856902143413
Greatest Common Divisor

GCD(a, b):

Largest integer \( d \) such that \( d \mid a \) and \( d \mid b \)

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

\[
\begin{align*}
2 & \cdot 3 & \cdot 5 \\
2 & \cdot 3 & \cdot 7 \\
3 & \cdot 3 & \cdot 22
\end{align*}
\]
GCD and Factoring

\[ a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46,200 \]
\[ b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204,750 \]

\[ \text{GCD}(a, b) = 2^{\min(3, 1)} \cdot 3^{\min(1, 2)} \cdot 5^{\min(2, 3)} \cdot 7^{\min(1, 1)} \cdot 11^{\min(1, 0)} \cdot 13^{\min(0, 1)} \]
\[ = 2 \cdot 3 \cdot 5^2 \cdot 7 = 150 \cdot 2 \cdot 1050 = 1050 \]

Factoring is expensive!
Can we compute \( \text{GCD}(a,b) \) without factoring?
Useful GCD Fact

If \( a \) and \( b \) are positive integers, then
\[
gcd(a, b) = gcd(b, a \mod b)
\]

Proof:

Let \( e = gcd(ab) \):

\( e \mid b \) and \( e \mid (a \mod b) \)

\( e \) is a multiple of \( e \)

\( e \leq gcd(ab) \)

Let \( d = gcd(ab) \):

\( d \mid a \) and \( d \mid b \)

\( a \mod b = a - q \cdot b \)

\( d \mid (a \mod b) \)

\( d \leq gcd(rb, a \mod b) \)

For some \( q = a \div b \).
Useful GCD Fact

**If** $a$ and $b$ are positive integers, then $\text{gcd}(a,b) = \text{gcd}(b, a \mod b)$

**Proof:**

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

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

Therefore $(a \mod b) = a - qb = kd - qjd = (k - qj)d$.

So, $d \mid (a \mod b)$ and since $d \mid b$ we must have $d \leq \text{gcd}(b, a \mod b)$.

Now, let $e = \text{gcd}(b, a \mod b)$. Then $e \mid b$ and $e \mid (a \mod b)$ so $b = me$ and $(a \mod b) = ne$ for some integers $m$ and $n$.

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

So, $e \mid a$ and since $e \mid b$ we must have $e \leq \text{gcd}(a, b)$.

It follows that $\text{gcd}(a, b) = \text{gcd}(b, a \mod b)$. ■
Another simple GCD fact

If $a$ is a positive integer, $\text{gcd}(a,0) = a$. 
Euclid’s Algorithm

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

```c
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 \( \gcd(a, b) = \gcd(b, a \mod b) \) to reduce numbers until you get \( \gcd(g, 0) = g \).

\[
\begin{align*}
gcd(660, 126) &= \frac{660}{126} + 30 \\
126 &= 4 \cdot 30 + 6 \\
30 &= 5 \cdot 6 + 0
\end{align*}
\]
Euclid’s Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \mod b)$ 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 $s$ and $t$ such that

$$\gcd(a, b) = sa + tb.$$
Extended Euclidean algorithm

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

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

\[

gcd(35, 27) = gcd(27, 35 \mod 27) = gcd(27, 8) \quad (35 = 1 \times 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 Intermediary Information):**

\[
\begin{array}{ccccccc}
 a & b & b & a \mod b & b & r & a = q \cdot b + r \\
35 & 27 & & & & & (35 = 1 \cdot 27 + 8) \\
27 & 35 \mod 27 & = & 35 & 27 & 8 & (35 = 1 \cdot 27 + 8) \\
27 & 8 & = & 8 & 27 \mod 8 & 3 & (27 = 3 \cdot 8 + 3) \\
8 & 27 \mod 8 & = & 8 & 3 & 2 & (8 = 2 \cdot 3 + 2) \\
3 & 8 \mod 3 & = & 3 & 8 & 2 & (8 = 2 \cdot 3 + 2) \\
2 & 3 \mod 2 & = & 2 & 3 & 1 & (3 = 1 \cdot 2 + 1) \\
1 & 2 \mod 1 & = & 1 & 2 & 0 & (2 = 2 \cdot 1 + 0)
\end{array}
\]
Extended Euclidean algorithm

• Can use Euclid’s Algorithm to find $s, t$ such that

\[ \text{gcd}(a, b) = sa + tb \]

Step 2 (Solve the equations for $r$):

\[
\begin{align*}
  a &= q \times b + r \\
  35 &= 1 \times 27 + 8 \\
  27 &= 3 \times 8 + 3 \\
  8 &= 2 \times 3 + 2 \\
  3 &= 1 \times 2 + 1 \\
  2 &= 2 \times 1 + 0
\end{align*}
\]

\[
\begin{align*}
  r &= a - q \times b \\
  8 &= 35 - 1 \times 27
\end{align*}
\]
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 $r$):

\[
\begin{align*}
\text{a} &= q \times \text{b} + r \\
35 &= 1 \times 27 + 8 \\
27 &= 3 \times 8 + 3 \\
8 &= 2 \times 3 + 2 \\
3 &= 1 \times 2 + 1 \\
\end{align*}
\]

\[
\begin{align*}
\text{r} &= a - q \times \text{b} \\
8 &= 35 - 1 \times 27 \\
3 &= 27 - 3 \times 8 \\
2 &= 8 - 2 \times 3 \\
1 &= 3 - 1 \times 2 \\
\end{align*}
\]
Extended Euclidean algorithm

• Can use Euclid’s Algorithm to find $s, t$ such that
  \[ \gcd(a, b) = sa + tb \]

Step 3 (Backward Substitute Equations):

\[
\begin{align*}
8 &= 35 - 1 \times 27 \\
3 &= 27 - 3 \times 8 \\
2 &= 8 - 2 \times 3 \\
1 &= 3 - 1 \times 2
\end{align*}
\]

Plug in the def of 2

\[
\begin{align*}
1 &= 3 - 1 \times (8 - 2 \times 3) \\
&= 3 - 8 + 2 \times 3 \\
&= (-1) \times 8 + 3 \times 3
\end{align*}
\]

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):

1. $$8 = 35 - 1 \times 27$$
2. $$3 = 27 - 3 \times 8$$
3. $$2 = 8 - 2 \times 3$$
4. $$1 = 3 - 1 \times 2$$

Plug in the def of 2

Re-arrange into 3’s and 8’s

Plug in the def of 3

Re-arrange into 8’s and 27’s

Re-arrange into 27’s and 35’s

$$1 = 3 - 1 \times (8 - 2 \times 3)$$
$$= 3 - 8 + 2 \times 3$$
$$= (-1) \times 8 + 3 \times 3$$

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

$$= 3 \times 27 + (-10) \times (35 - 1 \times 27)$$
$$= 3 \times 27 + (-10) \times 35 + 10 \times 27$$

$$= 13 \times 27 + (-10) \times 35$$
Suppose \( \gcd(a, m) = 1 \)

By Bézout’s Theorem, there exist integers \( s \) and \( t \) such that \( sa + tm = 1 \).

\( s \mod m \) is the multiplicative inverse of \( a \):

\[ 1 = (sa + tm) \mod m = sa \mod m \]
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{align*}
26 &= 7 \cdot 3 + 5 \\
7 &= 5 \cdot 1 + 2 \\
5 &= 2 \cdot 2 + 1
\end{align*}

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

Now \((-11) \mod 26 = 15\). 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

\[ \gcd(a, m) = 1 \text{ if } m \text{ is prime and } 0 < a < m \text{ so you can always solve these equations mod a prime.} \]

\[
\begin{array}{c|cccccc}
+ & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
0 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 0 \\
2 & 2 & 3 & 4 & 5 & 6 & 0 & 1 \\
3 & 3 & 4 & 5 & 6 & 0 & 1 & 2 \\
4 & 4 & 5 & 6 & 0 & 1 & 2 & 3 \\
5 & 5 & 6 & 0 & 1 & 2 & 3 & 4 \\
6 & 6 & 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[
\begin{array}{c|cccccc}
\times & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 0 & 2 & 4 & 6 & 1 & 3 & 5 \\
3 & 0 & 3 & 6 & 2 & 5 & 1 & 4 \\
4 & 0 & 4 & 1 & 5 & 2 & 6 & 3 \\
5 & 0 & 5 & 3 & 1 & 6 & 4 & 2 \\
6 & 0 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{array}
\]

mod 7