Foundations of Computer Science
Lecture 10

Number Theory

Division and the Greatest Common Divisor
Fundamental Theorem of Arithmetic
Cryptography and Modular Arithmetic
RSA: Public Key Cryptography
Last Time


2. Tools for summation: constant rule, sum rule, common sums and nested sum rule.

3. Comparing functions - asymptotics: Big-Oh, Theta, Little-Oh notation.
   \[ \log \log(n) < \log^{\alpha}(n) < n^\epsilon < 2^{\delta n} \]

   \[
   \sum_{i=1}^{n} i^k \sim \frac{n^{k+1}}{k+1} \quad \sum_{i=1}^{n} \frac{1}{i} \sim \ln n \quad \ln n! = \sum_{i=1}^{n} \ln i \sim n \ln n - n
   \]
Today: Number Theory

1 Division and Greatest Common Divisor (GCD)
   - Euclid’s algorithm
   - Bezout’s identity

2 Fundamental Theorem of Arithmetic

3 Modular Arithmetic
   - Cryptography
   - RSA public key cryptography
The Basics

Number theory has attracted the best of the best, because

“Babies can ask questions which grown-ups can’t solve” – P. Erdős

6 = 1 + 2 + 3 is perfect (equals the sum of its proper divisors). Is there an odd perfect number?

Quotient-Remainder Theorem

For $n \in \mathbb{Z}$ and $d \in \mathbb{N}$, $n = qd + r$. The quotient $q \in \mathbb{Z}$ and remainder $0 \leq r < d$ are unique.

e.g. $n = 27, d = 6$: $27 = 4 \cdot 6 + 4 \rightarrow \text{rem}(27, 6) = 4$.

Divisibility. $d$ divides $n$, $d|n$ if and only if $n = qd$ for some $q \in \mathbb{Z}$. e.g. $6|24$.

Primes. $P = \{2, 3, 5, 7, 11, \ldots\} = \{p \mid p \geq 2$ and the only positive divisors of $p$ are 1, $p\}$.

Division Facts (Exercise 10.2)

1. $d|0$.
2. If $d|m$ and $d'|n$, then $dd'|mn$.
3. If $d|m$ and $m|n$, then $d|n$.
4. If $d|n$ and $d|m$, then $d|n + m$.
5. If $d|n$, then $xd|xn$ for $x \in \mathbb{N}$.
6. If $d|m + n$ and $d|m$, then $d|n$. 
Greatest Common Divisor

Divisors of 30: \{1, 2, 3, 5, 6, 15, 30\}. Divisors of 42: \{1, 2, 3, 6, 7, 14, 21, 42\}. Common divisors: \{1, 2, 3, 6\}. 

*greatest common divisor (GCD) = 6.*

**Definition. Greatest Common Divisor, GCD**

Let \(m, n\) be two integers not both zero. \(\gcd(m, n)\) is the largest integer that divides both \(m\) and \(n\): \(\gcd(m, n)|m, \gcd(m, n)|n\) and any other common divisor \(d \leq \gcd(m, n)\).

Notice that every common divisor divides the GCD. Also, \(\gcd(m, n) = \gcd(n, m)\).

**Relatively Prime**

If \(\gcd(m, n) = 1\), then \(m, n\) are relatively prime.

Example: 6 and 35 are not prime but they are relatively prime.

**Theorem.**

\[\gcd(m, n) = \gcd(\text{rem}(n, m), m)\].

**Proof.** \(n = qm + r \rightarrow r = n - qm\). Let \(D = \gcd(m, n)\) and \(d = \gcd(m, r)\).

\(D|m\) and \(D|n \rightarrow D\) divides \(r = n - qm\). Hence, \(D \leq \gcd(m, r) = d\)  
\(d|m\) and \(d|r \rightarrow d\) divides \(n = qm + r\). Hence, \(d \leq \gcd(m, n) = D\). \(D \leq d\) and \(D \geq d \rightarrow D = d\), which proves \(\gcd(m, n) = \gcd(n, r)\).  

\((D\) is a common divisor of \(m, r)\)  
\((d\) is a common divisor of \(m, n)\)
Euclid’s Algorithm

**Theorem.**

\[ \gcd(m, n) = \gcd(\text{rem}(n, m), m). \]

\[
\begin{align*}
\gcd(42, 108) &= \gcd(24, 42) & 24 &= 108 - 2 \cdot 42 \\
&= \gcd(18, 24) & 18 &= 42 - 24 - (108 - 2 \cdot 42) = 3 \cdot 42 - 108 \\
&= \gcd(6, 18) & 6 &= 24 - 18 - (108 - 2 \cdot 42) - (3 \cdot 42 - 108) = 2 \cdot 108 - 5 \cdot 42 \\
&= \gcd(0, 6) & 0 &= 18 - 3 \cdot 6 \\
&= 6 & \gcd(0, n) &= n
\end{align*}
\]

Remainders in Euclid’s algorithm are integer linear combinations of 42 and 108.

In particular, \( \gcd(42, 108) = 6 = 2 \times 108 - 5 \times 42. \)

This will be true for \( \gcd(m, n) \) in general:

\[ \gcd(m, n) = mx + ny \quad \text{for some } x, y \in \mathbb{Z}. \]
Bezout’s Identity: A “Formula” for GCD

From Euclid’s Algorithm,
\[ \gcd(m, n) = mx + ny \quad \text{for some } x, y \in \mathbb{Z}. \]

Can any smaller positive number \( z \) be a linear combination of \( m \) and \( n \)?

suppose:
\[ z = mx + ny > 0. \]

\( \gcd(m, n) \) divides RHS \( \rightarrow \) \( \gcd(m, n) \mid z \), i.e \( z \geq \gcd(m, n) \) (because \( \gcd(m, n) \mid m \) and \( \gcd(m, n) \mid n \)).

**Theorem. Bezout’s Identity**
\( \gcd(m, n) \) is the *smallest positive integer linear combination* of \( m \) and \( n \):
\[ \gcd(m, n) = mx + ny \quad \text{for } x, y \in \mathbb{Z}. \]

**Formal Proof.** Let \( \ell \) be the smallest positive linear combination of \( m, n \): \( \ell = mx + ny \).
- Prove \( \ell \geq \gcd(m, n) \) as above.
- Prove \( \ell \leq \gcd(m, n) \) by showing \( \ell \) is a common divisor (\( \rem(m, \ell) = \rem(n, \ell) = 0 \)).

There is no “formula” for GCD. But this is close to a “formula”.
proof.

1. gcd(m, n) = mx + ny. Any common divisor divides the RHS and so also the LHS.
   (e.g. 1,2,3,6 are common divisors of 30,42 and all divide the GCD 6)

2. gcd(km, kn) = kmx + kny = k(mx + ny). The RHS is the smallest possible, so there is no smaller positive linear combination of m, n. That is gcd(m, n) = (mx + ny).
   (e.g. gcd(6, 15) = 3 -> gcd(12, 30) = 2 × 3 = 6)

3. 1 = ℓx + my and 1 = ℓx' + ny'. Multiplying,
   1 = (ℓx + my)(ℓx' + ny') = ℓ · (ℓxx' + nxy' + myx') + mn · (yy').
   (e.g. gcd(15, 4) = 1 and gcd(15, 7) = 1 -> gcd(15, 28) = 1)

4. dx + my = 1 → ndx + nmy = n. Since d|mn, d divides the LHS, hence d|n, the RHS.
   (e.g. gcd(4, 15) = 1 and 4|15 × 16 → 4|16)
Given 3 and 5-gallon jugs, measure exactly 4 gallons.

1: Repeatedly fill the 3-gallon jug.
2: Empty the 3-gallon jug into the 5-gallon jug.
3: If ever the 5-gallon jug is full, empty it by discarding the water.

\[
(0, 0) \xrightarrow{1} (3, 0) \xrightarrow{2} (0, 3) \xrightarrow{1} (3, 3) \xrightarrow{2} (1, 5) \xrightarrow{3} (1, 0) \xrightarrow{2} (0, 1) \xrightarrow{1} (3, 1) \xrightarrow{2} (0, 4)
\]

After the 3-gallon jug is emptied into the 5-gallon jug, the state is \((0, \ell)\), where

\[
\ell = 3x - 5y.
\]

(integer linear combination of 3, 5). Since \(\gcd(3, 5) = 1\) we can get \(\ell = 1\),

\[
1 = 3 \cdot 2 - 5 \cdot 1
\]

(after emptying the 3-gallon jug 2 times and the 5-gallon jug once, there is 1 gallon)

Do this 4 times and you have 4 gallons (guaranteed). (Actually fewer pours works.)

\[
(0, 0) \xrightarrow{1} (3, 0) \xrightarrow{2} (0, 3) \xrightarrow{1} (3, 3) \xrightarrow{2} (1, 5) \xrightarrow{3} (1, 0) \xrightarrow{2} (0, 1)
\]

(repeat 4 times)

If the producers of Die Hard had chosen 3 and 6 gallon jugs, there can be no sequel (phew 😊). (Why?)
Theorem. Uniqueness of Prime Factorization
Every \( n \geq 2 \) is uniquely (up to reordering) a product of primes.

**Euclid’s Lemma:** For primes \( p, q_1, \ldots, q_\ell \), if \( p | q_1 q_2 \cdots q_\ell \) then \( p \) is one of the \( q_i \).

Proof of lemma: If \( p | q_\ell \) then \( p = q_\ell \). If not, \( \gcd(p, q_\ell) = 1 \) and \( p | q_1 \cdots q_{\ell-1} \) by GCD fact (v). Induction on \( \ell \).

**Proof.** (FTA) Contradiction. Let \( n_* \) be the smallest counter-example, \( n_* > 2 \) and
\[
 n_* = p_1 p_2 \cdots p_n \\
= q_1 q_2 \cdots q_k
\]
Since \( p_1 | n_* \), it means \( p_1 | q_1 q_2 \cdots q_k \) and by Euclid’s Lemma, \( p_1 = q_i \) (w.l.o.g. \( q_1 \)).
\[
 n_* / p_1 = p_2 \cdots p_n \\\n= q_2 \cdots q_k.
\]
That is, \( n_* / p_1 \) is a smaller counter-example. **FISHY!**
Cryptography 101: Alice and Bob wish to securely exchange the prime $M$

Alice encrypts $M$ to $M_*$ and sends $M_*$ to Bob.
Bob decrypts $M_*$ to $M'$.
Charlie eavesdrops.

Example.
Alice Encrypts: $M_* = M \times k$
Alice and Bob know $k$, Charlie does not.
Bob Decrypts: $M' = M_* / k = M \times k / k = M$.

(Hooray, $M' = M$ and Charlie is in the dark.)

Secure as long as Charlie cannot factor $M'$ into $k$ and $M$.
One time use. For two cypher-texts, $k = \text{gcd}(M_1*, M_2*)$.
To improve, we need modular arithmetic.

Factoring is HARD.
Modular Arithmetic

\[ a \equiv b \pmod{d} \quad \text{if and only if} \quad d|(a - b), \quad \text{i.e.} \ a - b = kd \text{ for } k \in \mathbb{Z} \]

\[ 41 \equiv 79 \pmod{19} \quad \text{because} \quad 41 - 79 = -38 = -2 \cdot 19. \]

**Modular Equivalence Properties.**

Suppose \( a \equiv b \pmod{d} \), i.e. \( a = b + kd \), and \( r \equiv s \pmod{d} \), i.e. \( r = s + \ell d \). Then,

(a) \( ar \equiv bs \pmod{d} \).

(b) \( a + r \equiv b + s \pmod{d} \).

(c) \( a^n \equiv b^n \pmod{d} \).

\[
\begin{align*}
ar - bs &= (b + kd)(s + \ell d) - bs \\
&= d(ks + b\ell + k\ell d).
\end{align*}
\]

That is \( d|ar - bs \).

\[
\begin{align*}
(a + r) - (b + s) &= (b + kd + s + \ell d) - b - s \\
&= d(k + \ell).
\end{align*}
\]

That is \( d|(a + r) - (b + s) \).

Repeated application of (a) Induction.

Addition and multiplication are just like regular arithmetic.

**Example.** What is the last digit of \( 3^{2017} \)?

\[
\begin{align*}
3^2 &\equiv -1 \pmod{10} \\
\Rightarrow (3^2)^{1008} &\equiv (-1)^{1008} \pmod{10} \\
\Rightarrow 3 \cdot (3^2)^{1008} &\equiv 3 \cdot (-1)^{1008} \pmod{10} \\
&\equiv 3
\end{align*}
\]
Modular Division is Not Like Regular Arithmetic

\[ 15 \cdot 6 \equiv 13 \cdot 6 \quad (\text{mod } 12) \]
\[ 15 \not\equiv 13 \quad (\text{mod } 12) \]

\[ 15 \cdot 6 \equiv 2 \cdot 6 \quad (\text{mod } 13) \]
\[ 15 \equiv 2 \quad (\text{mod } 13) \]

\[ 7 \cdot 8 \equiv 22 \cdot 8 \quad (\text{mod } 15) \]
\[ 7 \equiv 22 \quad (\text{mod } 15) \]

Modular Division: cancelling a factor from both sides

Suppose \( ac \equiv bc \ (\text{mod } d) \). You can cancel \( c \) to get \( a \equiv b \ (\text{mod } d) \) if \( \gcd(c, d) = 1 \).

**Proof.** \( d|c(a - b) \). By GCD fact (v), \( d|a - b \) because \( \gcd(c, d) = 1 \).

If \( d \) is prime, then division with prime modulus is pretty much like regular division.

**Modular Inverse.** Inverses do not exist in \( \mathbb{N} \). Modular inverse may exist.

\[ 3 \times n = 1 \quad n =? \]
\[ 3 \times n = 1 \ (\text{mod } 7) \quad n = 5 \]
RSA Public Key Cryptography Uses Modular Arithmetic

Bob broadcasts to the world the numbers 23, 55. (Bob’s RSA public key).

\[ M \rightarrow M_* \equiv M^{23} \pmod{55} \rightarrow M' \equiv M_*^7 \pmod{55} \]

Examples. Does Bob always decode to the correct message?

**Exercise 10.14.** Proof that Bob always decodes to the right message for special 55, 23 and 7. (How to get them?)

**Practical Implementation.** Good idea to pad with random bits to make the cypher text random.