Integers and Division

Notations

$\mathcal{Z}$: set of integers
$\mathcal{N}$: set of natural numbers
$\mathcal{R}$: set of real numbers
$\mathcal{Z}^+$: set of positive integers

Some elements of number theory are needed in:

Data structures,
Random number generation,
Encryption of data for secure data
transmission,
Scheduling, etc.

Definition: For integers a and b with $a \neq 0$ we define

a **divides** b iff $\exists$ an integer c such that

$$b = ac$$

a divides b is written as $a \mid b$

$3 \mid 15$
$3 \nmid 16$
$4 \mid 16$
$16 \nmid 4$

$a \neq 0$ and $a \mid b$ is equivalent to each of:

a is a **factor** of b

b is a **multiple** of a

Theorem: Let a, b, and c be integers. Then

(1) if $a \mid b$ and $a \mid c$ then $a \mid (b + c)$.

(2) if $a \mid b$ then $a \mid bc$ for all integers c.

(3) if $a \mid b$ and $b \mid c$ then $a \mid c$.

Prime and composite numbers

A **prime** is a positive integer p that has only two distinct positive factors, 1 and p.

Examples: $2, 3, 5, 7, 11, 13, 29, 53, 997, 7951, \ldots$

A positive integer greater that 1 which is not a prime is called **composite**.

Examples: $6 = 2 \cdot 3$, $35 = 5 \cdot 7$, $57 = 3 \cdot 19$, etc.

Fundamental Theorem of Arithmetic *Every positive integer* $n \geq 2$ *can be written uniquely as a product of primes.*

Proof (by strong induction).

Basis. $n = 2$ can be written as a trivial product of primes.

Induction hypothesis. Assume that any integer $2 \leq k < n$ can we written as a product of primes.

Induction step. If n is prime we are done. If n is not a prime it is composite, i.e., $n = n_1 n_2$, where $2 \leq n_1, n_2 < n$. By induction hypothesis n_1 and n_2 can be factored into product of primes so can be n.

Large primes are used in *cryptology*.

$$40 = 2 \cdot 2 \cdot 2 \cdot 5 = 2^3 \cdot 5$$

$$42 = 2 \cdot 3 \cdot 7$$

$$780 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 13 = 2^2 \cdot 3 \cdot 5 \cdot 13$$

$$550 = 2 \cdot 5 \cdot 5 \cdot 11 = 2 \cdot 5^2 \cdot 11$$

Theorem *If* n *is a composite number then* n *has a prime factor* $\leq \sqrt{n}$.

Proof. If n is composite then n has a factor $a, 1 < a < n$, hence $n = ab, a, b > 1$. So $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$ (otherwise $ab > n$). Assume without loss of generality that $a \leq \sqrt{n}$. Then either a is prime or it has a prime factor less than $a \leq \sqrt{n}$.

This is an important bound when trying to find a factorization of a number.

Example 1: $n = 311$

$\sqrt{311} \doteq 17.6$

Test division by 2, 3, 5, 7, 11, 13, 17.

If none of these divides 311, it is a prime,
otherwise we have found a factor. 311 is a prime number.

Example 2: $n = 253$

$\sqrt{253} \doteq 15.9$

Test division by 2, 3, 5, 7, 11, 13.

253 = 11*23 so 253 is composite.

Factorization of very large numbers by computers is a difficult problem.

This fact is used by some encryption systems.
RSA encryption system, named after the inventors Rivest, Shamir, and Adelman.

Breaking a code would require factoring numbers with 250 to 500 digits that have only two prime factors, both large primes.

The Division Algorithm

Let a be an integer and d a positive integer. Then there exist unique integers q and r, $0 \leq r < d$, such that

$$a = dq + r$$

a is called the **dividend**
d is called the **divisor**
r is called the **remainder**
q is called the **quotient**.

GCD and LCM

Definition: $GCD(a, b)$, called the **greatest common divisor** of a and b, is the largest factor of a and b.

$GCD(18, 24) = 6$
$GCD(18, 13) = 1$

When $GCD(a, b) = 1$, we say that a and b are relatively prime (or coprime)

Definition: $LCM(a, b)$ is the **least common multiple** of a and b. It is the smallest integer having a and b as factors.

$LCM(8, 6) = 24$
$LCM(8, 12) = 24$
$LCM(11, 17) = 11 \cdot 17 = 187$

GCD and LCM

The prime factorization of a and b can be used to find $GCD(a,b)$ or $LCM(a,b)$:

$$780 = 2 \cdot 2 \cdot 3 \cdot 5 \cdot 13 = 2^2 \cdot 3 \cdot 5 \quad \cdot \quad 13$$
$$550 = 2 \cdot 5 \cdot 5 \cdot 11 \quad = 2 \quad \cdot \quad 5^2 \cdot 11$$

$GCD(780, 550) = 2 \cdot 5 = 10$
take the factors common to both numbers with the lowest exponent.

$LCM(780, 550) = 2^2 \cdot 3 \cdot 5^2 \cdot 11 \cdot 13 = 42900$
take all factors in both numbers with the highest exponent.

If $a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$ and
$b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n}$

$$gcd(a,b) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \cdots p_n^{min(a_n,b_n)}$$

$$lcm(a,b) = p_1^{max(a_1,b_1)} p_2^{max(a_2,b_2)} \cdots p_n^{max(a_n,b_n)}$$

Note that $min(a_i,b_i) + max(a_i,b_i) = a_i + b_i$, leading to

---

Theorem

Let a and b be positive integers. Then

$$ab = gcd(a,b) \cdot lcm(a,b)$$

Example:
$GCD(780, 550) = 2 \cdot 5 = 10$
$780 \cdot 550 = 429000$
$LCM(780, 550) = 42900$

Co-prime integers

Definition: The integers a and b are said to be **co-prime** or **relatively prime** if $gcd(a, b) = 1$.

Example 1:
6 and 25 are co-prime, as $gcd(6, 25) = 1$.

Example 2:
6 and 27 are not co-prime, since $gcd(6, 27) = 3 \neq 1$.

Example 3:
Any two distinct prime numbers are relatively prime.

Modular Arithmetic

Let a be an integer and m be a positive integer.

$$a \bmod m$$

is defined as the remainder when a is divided by m.

$$0 \leq (a \bmod m) < m$$

$8 \bmod 7 = 1$
$12 \bmod 7 = 5$
$30 \bmod 7 = 2$
$51 \bmod 7 = 2$
$21 \bmod 7 = 0$

Since the result of the *mod* operation must be ≥ 0 and < 7,

$-3 \bmod 7 = 4$ since $-3 = -1 \cdot 7 + 4$
$-22 \bmod 6 = 2$ since $-22 = -4 \cdot 6 + 2$

Example of the use of *mod*:

A scheduling problem:

We have *processors* $1, 2, 3, 4, 5$
and *jobs* $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, \ldots$

Scheduling: Given a job number, select a processor on which to execute the job.

round-robin scheduling:

jobs $1, 6, 11, 16, 21, \ldots$ are done on processor 2
jobs $2, 7, 12, 17, 22, \ldots$ are done on processor 3
jobs $3, 8, 13, 18, 23, \ldots$ are done on processor 4
jobs $4, 9, 14, 19, 24, \ldots$ are done on processor 5
jobs $5, 10, 15, 20, 25, \ldots$ are done on processor 1

job i is assigned to processor $(i \bmod 5) + 1$

Congruences

Definition: Let a and b be integers and m be a positive integer. We say that
a is **congruent** to b **modulo** m if $m \mid (a - b)$.

$$a \equiv b \pmod{m}$$

Examples:

$5 \mid (14 - 9) \quad \Leftrightarrow \quad 14 \equiv 9 \pmod{5}$

$5 \mid (19 - 9) \quad \Leftrightarrow \quad 19 \equiv 9 \pmod{5}$

$5 \mid (32 - 12) \quad \Leftrightarrow \quad 32 \equiv 12 \pmod{5}$

$7 \mid (14 - 7) \quad \Leftrightarrow \quad 14 \equiv 7 \pmod{7}$

Theorem

Let a and b be integers and m be a positive integer.

$a \equiv b \pmod{m} \quad \Leftrightarrow \quad (a \bmod m) = (b \bmod m)$

Theorem

Let a and b be integers and m be a positive integer.

$$a \equiv b \pmod{m} \text{ iff } a = b + km \text{ for some integer } k$$

Problem:

Find all integers congruent to 7 modulo 6.

It is the infinite set $\{a : a = 7 + 6k,\ k \in Z\}$.

$7 \equiv 13 \pmod{6}$	$7 \equiv 19 \pmod{6}$
$7 \equiv 25 \pmod{6}$	$7 \equiv 31 \pmod{6}$
$7 \equiv 37 \pmod{6}$	$7 \equiv 1 \pmod{6}$
$7 \equiv -5 \pmod{6}$	$7 \equiv -11 \pmod{6}$

Theorem.

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then

$$a + c \equiv b + d \pmod{m}$$

$$a \cdot c \equiv b \cdot d \pmod{m}$$

Applications

Hashing Functions

Assign memory locations to files/records so that they can be retrieved quickly.

Records like student records are identified by a **key**, which uniquely identifies each record.

Hashing function h assigns memory location $h(k)$ to the record that has k as its key.

One of the hashing functions often used is:

$$h(k) = k \pmod{m}$$

where m is the number of available memory locations.

Hashing function should be onto so that all memory locations are possible, but it is not one-to-one (there are more possible keys than memory locations.) When this happens more than one file may be assigned to a memory location, we say that a collision occurs.

Pseudorandom numbers: Choose 4 integers:
m - the modulus,
a - the multiplier,
c - the increment,
x_0 - the seed.
$2 \leq a < m$ and $0 \leq c,\ x_0 < m$

$$x_{n+1} \equiv (ax_n + c) \bmod m$$

$n = 0, 1, 2, \ldots.$

Cryptology: Primitive encryption is to shift each letter in the English alphabet by m positions forward (or backward).

Example: In the English alphabet, each letter from a to z is assigned an integer from 0 to 25 respectively. A letter in position p is encrypted by:

$$f(p) = (p + m) \text{ mod } 26$$

To recover the message, do f^{-1}:

$$f^{-1}(p) = (p - m) \text{ mod } 26$$

Obviously this method does not provide a high level of security.