Let us get into…

Number Theory

(chapter 4)

Introduction to Number Theory

Number theory is about **integers** and their properties.

We will start with the basic principles of

- divisibility,
- greatest common divisors,
- least common multiples, and
- modular arithmetic

and look at some relevant algorithms.

Division

If a and b are integers with a ≠ 0, we say that a ***divides*** b if there is an integer c so that b = ac.

When a divides b we say that a is a ***factor*** of b and that b is a ***multiple*** of a.

The notation ***a | b*** means that a divides b.

We write ***a X b*** when a does not divide b (see book for correct symbol).

Divisibility Theorems (Th. 1, p. 238)

For integers a, b, and c it is true that

- if a | b and a | c, then a | (b + c)
  Example: 3 | 6 and 3 | 9, so 3 | 15.
- if a | b, then a | bc for all integers c
  Example: 5 | 10, so 5 | 20, 5 | 30, 5 | 40, …
- if a | b and b | c, then a | c
  Example: 4 | 8 and 8 | 24, so 4 | 24.

Proof of Theorem 1, p. 238

- If a | b and a | c, then a | (b + c)

Proof: a | b means b = au for some integer u.

b = au and c = av, where u and v are integers.

Then b+c = au + av = a(u+v), so a | (b + c)

Proof continued.

- If a | b, then a | bc for all integers c.
  proof: b = au, so bc = auc, so a | bc.
- If a | b and b | c, then a | c
  proof: b = au, c = bv, so c = auv, and so a | c.

Corollary 1, p. 239

If a, b and c are integers such that a | b and a | c then a | mb+nc, where m and n are integers.

Proof:

This follows directly from Theorem 1.

The Division Algorithm (Th. 2, p 239)

Let **a** be an integer and **d** a positive integer. Then there are unique integers **q** and **r**, with **$0 \le r < d$**, such that **a = dq + r**.

In the above equation,

- **d** is called the *divisor*,
- **a** is called the *dividend*,
- **q** is called the *quotient*, we say q = a div d, and
- **r** is called the *remainder*. We say r = a mod d
  (See Def. 2, page 239)

The Division Algorithm

Example:

When we divide 17 by 5, we have

$17 = 5 \cdot 3 + 2$.

- 17 is the dividend,
- 5 is the divisor,
- 3 is the quotient, and
- 2 is the remainder.

The Division Algorithm

Another example:

What happens when we divide -11 by 3 ?

Note that the remainder cannot be negative.

$-11 = 3 \cdot (-4) + 1$.

- -11 is the dividend,
- 3 is the divisor,
- -4 is the quotient, and
- 1 is the remainder.

The Division Algorithm

Example:

When we divide 21 by 5, we have

$21 = 5 \cdot 4 + 1$.

- 21 is the dividend,
- 5 is the divisor,
- 4 is called the quotient, and
- 1 is called the remainder.

Proof of the Division Algorithm

Given integers a, d>0, ∃ unique q,r such that a = dq + r, and $0 \le r < d$.

Proof. To see this consider the set of all multiples of d on the number line.

Each integer a can be written uniquely as dq +r, where dq is a, or the multiple of d to the immediate left of a.

Clinching the uniqueness

Suppose $a = dq_1 + r_1,\ 0 \le r_1 < d$, and
$a = dq_2 + r_2,\ 0 \le r_2 < d$.

Then subtracting we get
$0 = d(q_1 - q_2) + (r_1 - r_2)$

Then $d \mid (r_1 - r_2)$ and $-d < (r_1 - r_2) < d$,
so $(r_1 - r_2) = 0$
and hence $q_1 = q_2$.

This proves uniqueness of q and r.

Primes

A positive 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.

The Fundamental Theorem of Arithmetic:
(p. 258)

Every positive integer bigger than 1 can be written **uniquely** as the **product of primes**, where the prime factors are written in order of increasing size.
(proof later…)

Primes

Examples:

15 =	$3 \cdot 5$
48 =	$2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3$
17 =	17
100 =	$2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
512 =	$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^9$
515 =	$5 \cdot 103$
28 =	$2 \cdot 2 \cdot 7 = 2^2 \cdot 7$

Theorem 2, p. 258

If n is a composite integer then n has a prime factor $\le \sqrt{n}$. ($\le$ sqrt(n))

Proof: If n is composite then $n = uv$, where one of u and v must be $\le \sqrt{n}$.

This factor $\le \sqrt{n}$ then must have a prime factor also $\le \sqrt{n}$.

Infinitely many primes…

Theorem: There are infinitely many primes.

Proof: Suppose there are only n primes, $p_1, p_2, \ldots, p_n$.

Then $u = p_1 p_2 \ldots p_n + 1$ has a prime divisor but it can't be one of $p_1, p_2, \ldots, p_n$.

Greatest Common Divisors

Let a and b be integers, not both zero.
The largest integer d such that $d \mid a$ and $d \mid b$ is called the **greatest common divisor** of a and b.

The greatest common divisor of a and b is denoted by gcd(a, b).

Example 1: What is gcd(48, 72) ?

The positive common divisors of 48 and 72 are
1, 2, 3, 4, 6, 8, 12, 16, and 24, so gcd(48, 72) = 24.

Example 2: What is gcd(19, 72) ?

The only positive common divisor of 19 and 72 is 1, so gcd(19, 72) = 1.

Greatest Common Divisors

Using prime factorizations:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,
where $p_1 < p_2 < \ldots < p_n$ and $a_i, b_i \in \mathbf{N}$ for $1 \le i \le n$

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

Example:

$a = 60 = 2^2\, 3^1\, 5^1$

$b = 54 = 2^1\, 3^3\, 5^0$

$\gcd(a, b) = 2^1\, 3^1\, 5^0 = 6$

Relatively Prime Integers

Definition:

Two integers a and b are **relatively prime** if
gcd(a, b) = 1.
This means that no prime divides both a and b.

Examples:

Are 15 and 28 relatively prime?
Yes, gcd(15, 28) = 1.
Are 55 and 28 relatively prime?
Yes, gcd(55, 28) = 1.
Are 35 and 28 relatively prime?
No, gcd(35, 28) = 7.

Relatively Prime Integers

Definition:

The integers $a_1, a_2, \ldots, a_n$ are **pairwise relatively prime** if $\gcd(a_i, a_j) = 1$ whenever $1 \le i < j \le n$.

Examples:

Are 15, 17, and 27 pairwise relatively prime?
No, because gcd(15, 27) = 3.

Are 15, 17, and 28 pairwise relatively prime?
Yes, because gcd(15, 17) = 1, gcd(15, 28) = 1 and gcd(17, 28) = 1.

Least Common Multiples

Definition:

The **least common multiple** of the positive integers a and b is the smallest positive integer that is divisible by both a and b.

We denote the least common multiple of a and b by lcm(a, b).

Examples:

lcm(3, 7) = 21

lcm(4, 6) = 12

lcm(5, 10) = 10

Least Common Multiples

Using prime factorizations:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,
where $p_1 < p_2 < \ldots < p_n$ and $a_i, b_i \in \mathbf{N}$ for $1 \le i \le n$

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

Example:

$a = 60 = 2^2\, 3^1\, 5^1$

$b = 54 = 2^1\, 3^3\, 5^0$

$\text{lcm}(a, b) = 2^2\, 3^3\, 5^1 = 4 \cdot 27 \cdot 5 = 540$

GCD and LCM

$a = 60 = 2^2\, 3^1\, 5^1$

$b = 54 = 2^1\, 3^3\, 5^0$

$\gcd(a, b) = 2^1\, 3^1\, 5^0 = 6$

$\text{lcm}(a, b) = 2^2\, 3^3\, 5^1 = 540$

Theorem: $a \cdot b = \gcd(a,b) \cdot \text{lcm}(a,b)$

Th. a·b = gcd(a,b)·lcm(a,b)

Proof. Express a and b as products of primes.

If a prime p occurs with power i in a and power j in b, and i <= j then

p occurs with power i in gcd(a,b) and power j in lcm(a,b), thus with power i+j in the products a · b and gcd(a,b) · lcm(a,b)

This gives our theorem, since it holds for each such prime p.

Modular Arithmetic

Let a be an integer and m be a positive integer. We denote by **a mod m** the remainder when a is divided by m.

Examples:

9 mod 4 = 1

9 mod 3 = 0

9 mod 10 = 9

-13 mod 4 = 3

Congruences

Let a and b be integers and m be a positive integer. We say that **a is congruent to b modulo m** if m divides a – b.

We use the notation **a ≡ b (mod m)** to indicate that a is congruent to b modulo m.

In other words (Th. 3, page 241):
a ≡ b (mod m) if and only if **a mod m = b mod m**.

Congruences

Examples:

Is it true that 46 ≡ 68 (mod 11) ?
Yes, because 11 | (46 – 68).

Is it true that 46 ≡ 68 (mod 22)?
Yes, because 22 | (46 – 68).

For which integers z is it true that z ≡ 12 (mod 10)?
It is true for any z∈{…,-28, -18, -8, 2, 12, 22, 32, …}

Theorem (Th. 4, p. 241): Let m be a positive integer. The integers a and b are congruent modulo m if and only if there is an integer k such that a = b + km.

Congruences

Theorem (Th. 5, p. 242): Let m be a positive integer. If a ≡ b (mod m) and c ≡ d (mod m), then
a + c ≡ b + d (mod m) and ac ≡ bd (mod m).

Proof:
We know that a ≡ b (mod m) and c ≡ d (mod m) implies that there are integers s and t with
b = a + sm and d = c + tm.

Therefore,
b + d = (a + sm) + (c + tm) = (a + c) + m(s + t) and
bd = (a + sm)(c + tm) = ac + m(at + cs + stm).

Hence, a + c ≡ b + d (mod m) and ac ≡ bd (mod m).

A useful theorem on gcd

Theorem: if a = bq + r (a,b,q,r, are integers) then gcd(a,b) = gcd(b,r)

Proof: An integer x divides both a and b iff it divides both b and r. (Do you see why? Do you see the symmetry in the roles of a and r?)

Hence (a,b) and (b,r) have the same set of common divisors.

Hence gcd(a,b) = gcd(b,r). This is Lemma 1, p. 268.

This theorem is the basis of the Euclidean Algorithm.

The Euclidean Algorithm

The **Euclidean Algorithm** finds the **greatest common divisor** of two integers a and b.

For example, if we want to find gcd(287, 91), we **divide** 287 (the larger number) by 91 (the smaller one):

$287 = 91 \cdot 3 + 14$

Now, applying our previous Theorem, we see that
gcd(287, 91) = gcd(91, 14)

We have reduced the original problem to a smaller one.

The Euclidean Algorithm

gcd(287, 91) = gcd(91, 14).

We now divide 14 into 91:
$91 = 14 \cdot 6 + 7$

So we have
gcd(91, 14) = gcd(14,7).

We recognize that the answer is 7, but for the algorithm we have to continue, divide 7 into 14.

$14 = 7 \cdot 2 + 0$, so $7 \mid 14$.
Thus 7 = gcd(14, 7) = gcd(91, 14) = gcd(287, 91)

The Euclidean Algorithm

To summarize:

$287 = 91 \cdot 3 + 14$, so
gcd(287, 91) = gcd(91, 14)
$91 = 14 \cdot 6 + 7$,
gcd(91, 14) = gcd(14, 7)
$14 = 7 \cdot 2 + 0$,
$7 \mid 14$, so gcd(14, 7) = 7

Thus gcd(287, 91) = 7.

The Euclidean Algorithm

The **Euclidean Algorithm** finds the **greatest common divisor** of two integers a and b.

1. If $b < a$, divide b into a, get remainder r_1
   $a = bq_1+r_1$, $0 \le r_1 < b$. If $r_1=0$ we are done
   Now $\gcd(a,b) = \gcd(b,r_1)$. Repeat until remainder is 0.
2. $b = r_1q_2 + r_2$, $0 \le r_2 < r_1$. If $r_2=0$ we are done
   Now gcd(b,r1) = gcd(r1,r2).
3. $r_1 = r_2q_3 + r_3$, $0 \le r_3 < r_2$
   Now $\gcd(r_1,r_2) = \gcd(r_2,r_3)$.
4. Since the remainders are decreasing, we'll hit 0 after finitely many (very few, actually) steps.
5. When $r_n = r_{n+1}q_{n+1} + 0$, we have
   $r_{n+1} = \gcd(r_n, r_{n+1}) = \gcd(r_n, r_{n-1}) = \ldots = \gcd(a,b)$

The Euclidean Algorithm

In **pseudocode**, the algorithm can be implemented as follows:

```
procedure gcd(a, b: positive integers)
x := a
y := b
while y ≠ 0
begin
    r := x mod y
    x := y
    y := r
end
{x is gcd(a, b)}
```


Arithmetic Modulo *m*

Definitions: Let $\mathbf{Z}_m$ be the set of nonnegative integers less than *m*:
$\{0,1, \ldots, m-1\}$

The operation $+_m$ is defined as $a +_m b = (a + b) \textbf{ mod } m$. This is *addition modulo m*.

The operation $\cdot_m$ is defined as $a \cdot_m b = (a + b) \textbf{ mod } m$. This is *multiplication modulo m*.

Using these operations is said to be doing *arithmetic modulo m*.

Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definitions above:

- $7 +_{11} 9 = (7 + 9) \textbf{ mod } 11 = 16 \textbf{ mod } 11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9) \textbf{ mod } 11 = 63 \textbf{ mod } 11 = 8$

Arithmetic Modulo m

The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.

- *Closure*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbf{Z}_m$.
- *Associativity*: If a, b, and c belong to $\mathbf{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
- *Commutativity*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.
- *Identity elements*: The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.
  - If a belongs to $\mathbf{Z}_m$, then $a +_m 0 = a$ and $a \cdot_m 1 = a$.

Arithmetic Modulo m

- *Additive inverses*: If $a \neq 0$ belongs to $\mathbf{Z}_m$, then $m - a$ is the additive inverse of a modulo m and 0 is its own additive inverse.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- *Distributivity*: If a, b, and c belong to $\mathbf{Z}_m$, then
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

Exercises 42-44 ask for proofs of these properties.

Multiplicatative inverses have not been included since they do not always exist. For example, there is no multiplicative inverse of 2 modulo 6. But every non zero element of $\mathbf{Z}_m$ will have a multiplicative inverse if m is a prime.

(*optional*) Using the terminology of abstract algebra, $\mathbf{Z}_m$ with $+_m$ is a commutative group and $\mathbf{Z}_m$ with $+_m$ and $\cdot_m$ is a commutative ring.

If m is prime then $\mathbf{Z}_m$ is a field.

Representations of Integers

Let b be a positive integer greater than 1.
Then if n is a positive integer, it can be expressed **uniquely** in the form:

$n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$,

where k is a nonnegative integer,
$a_0, a_1, \ldots, a_k$ are nonnegative integers less than b, and $a_k \neq 0$.

Example for b=10:

$859 = 8 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0$

Representations of Integers

Example for b=2 (binary expansion):

$(10110)_2 = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 = (22)_{10}$

Example for b=16 (hexadecimal expansion):

(we use letters A to F to indicate numbers 10 to 15)

$(3A0F)_{16} = 3 \cdot 16^3 + 10 \cdot 16^2 + 0 \cdot 16^1 + 15 \cdot 16^0 = (14863)_{10}$

Representations of Integers

How can we construct the base b expansion of an integer n?

First, divide n by b to obtain a quotient q_0 and remainder a_0, that is,

$n = bq_0 + a_0$, where $0 \leq a_0 < b$.

The remainder a_0 is the rightmost digit in the base b expansion of n.

Next, divide q_0 by b to obtain:

$q_0 = bq_1 + a_1$, where $0 \leq a_1 < b$.

a_1 is the second digit from the right in the base b expansion of n. Continue this process until you obtain a quotient equal to zero.

Representations of Integers

Example:
What is the base 8 expansion of $(12345)_{10}$?

First, divide 12345 by 8:
$12345 = 8 \cdot 1543 + 1$

$1543 = 8 \cdot 192 + 7$
$192 = 8 \cdot 24 + 0$
$24 = 8 \cdot 3 + 0$
$3 = 8 \cdot 0 + 3$

The result is: $(12345)_{10} = (30071)_8$.

Representations of Integers

procedure base_b_expansion(n, b: positive integers)
q := n
k := 0
while q ≠ 0
begin
    a_k := q mod b
    q := $\lfloor q/b \rfloor$
    k := k + 1
end
{the base b expansion of n is $(a_{k-1} \ldots a_1 a_0)_b$ }

Addition of Integers

How do we (humans) add two integers?

Example:

```
  111      carry
  7583
+ 4932
------
 12515
```

Binary expansions:

$$\begin{array}{r} 1\ \ 1 \quad \text{carry} \\ (1011)_2 \\ +\ (1010)_2 \\ \hline (10101)_2 \end{array}$$

Addition of Integers

Let $a = (a_{n-1}a_{n-2}\ldots a_1a_0)_2$, $b = (b_{n-1}b_{n-2}\ldots b_1b_0)_2$.

How can we **algorithmically** add these two binary numbers?

First, add their rightmost bits:

$a_0 + b_0 = c_0 \cdot 2 + s_0$,

where s_0 is the **rightmost bit** in the binary expansion of a + b, and c_0 is the **carry**.

Then, add the next pair of bits and the carry:

$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1$,

where s_1 is the **next bit** in the binary expansion of a + b, and c_1 is the carry.

Addition of Integers

Continue this process until you obtain c_{n-1}.

The leading bit of the sum is $s_n = c_{n-1}$.

The result is:

$a + b = (s_n s_{n-1} \ldots s_1 s_0)_2$

Addition of Integers

Example:
Add $a = (1110)_2$ and $b = (1011)_2$.

$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1$, so that $c_0 = 0$ and $s_0 = 1$.

$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0$, so $c_1 = 1$ and $s_1 = 0$.

$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 \cdot 2 + 0$, so $c_2 = 1$ and $s_2 = 0$.

$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 \cdot 2 + 1$, so $c_3 = 1$ and $s_3 = 1$.

$s_4 = c_3 = 1$.

Therefore, $s = a + b = (11001)_2$.

Addition of Integers

procedure add(a, b: positive integers)
  // a_i, b_i are the bits of a and b.
c := 0
for j := 0 to n-1
begin
    d := $\lfloor (a_j + b_j + c)/2 \rfloor$ // gives the high bit of sum
    s_j := $a_j + b_j + c - 2d$ // gives the low bit of sum
    c := d
end
s_n := c
{the binary expansion of the sum is $(s_n s_{n-1} \ldots s_1 s_0)_2$}

Multiplication of Integers

procedure multiply(a, b: positive integers)
// a_i, b_i are the bits of a and b.
for j := 0 to n-1
begin
    if b_j = 1 then c_j := a shifted left j places
    else c_j := 0 // c_j are the partial products
end
p := 0
for i := 0 to n-1
    p := p + c_j
{p is the value of the product as an integer.
Note that we haven't computed bits for p}

More Algorithms

Take a look at Algorithms 4 and 5 on pages 253, 254 and be sure you understand them. It's important to be able to read the code and see what it says.

Algorithm 4 gives a way of doing the division algorithm using repeated subtractions instead of division.

Algorithm 5 gives a way of computing b^n using a binary representation of n