Let us get into…

Number Theory

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 $a \neq 0$, 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 **$a \mid b$** means that a divides b.

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

Divisibility Theorems

- 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.

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:
- Every positive integer can be written **uniquely** as the **product of primes**, where the prime factors are written in order of increasing size.

Primes

- Examples:

$15 = 3 \cdot 5$

$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3$

$17 = 17$

$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$

$512 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^9$

$515 = 5 \cdot 103$

$28 = 2 \cdot 2 \cdot 7$

Primes

- If n is a composite integer, then n has a prime divisor less than or equal $\sqrt{n}$

- This is easy to see: if n is a composite integer, it must have two prime divisors p_1 and p_2 such that $p_1 \cdot p_2 = n$.

- p_1 and p_2 cannot both be greater than
- $\sqrt{n}$, because then $p_1 \cdot p_2 > n$.

The Division Algorithm

- 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, and
- **r** is called the remainder.

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 called the quotient, and
- 2 is called 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 called the quotient, and
- 1 is called the remainder.

Greatest Common Divisors

- Let a and b be integers, not both zero.
- The largest integer d such that d | a and d | 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:**

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

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

- **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.
- **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 $1 \leq i < j \leq n$.
- **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 $1 \leq i \leq n$
- $\mathrm{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \ldots p_n^{\max(a_n, b_n)}$
- **Example:**

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

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

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

GCD and LCM

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

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

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

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

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

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 **$a \equiv b \pmod{m}$** to indicate that a is congruent to b modulo m.

- In other words:
  $a \equiv b \pmod{m}$ if and only if **$a \bmod m = b \bmod m$**.

Congruences

- **Examples:**
- Is it true that $46 \equiv 68 \pmod{11}$?
- Yes, because $11 \mid (46 - 68)$.
- Is it true that $46 \equiv 68 \pmod{22}$?
- Yes, because $22 \mid (46 - 68)$.
- For which integers z is it true that $z \equiv 12 \pmod{10}$?
- It is true for any $z \in \{\ldots, -28, -18, -8, 2, 12, 22, 32, \ldots\}$

- **Theorem:** 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:** Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.
- **Proof:**
- We know that $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ 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, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

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 by 91:
- $287 = 91 \cdot 3 + 14$
- We know that for integers a, b and c, if $a \mid b$ and $a \mid c$, then $a \mid (b + c)$.
- Therefore, any divisor of 287 and 91 must also be a divisor of $287 - 91 \cdot 3 = 14$.
- Consequently, gcd(287, 91) = gcd(14, 91).

The Euclidean Algorithm

- In the next step, we divide 91 by 14:
- $91 = 14 \cdot 6 + 7$
- This means that gcd(14, 91) = gcd(14, 7).
- So we divide 14 by 7:
- $14 = 7 \cdot 2 + 0$
- We find that $7 \mid 14$, and thus gcd(14, 7) = 7.
- **Therefore, gcd(287, 91) = 7.**

The Euclidean Algorithm

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

- **procedure** gcd(a, b: positive integers)
- x := a
- y := b
- **while** y $\neq$ 0
- **begin**
- r := x **mod** y
- x := y
- y := r
- **end** {x is gcd(a, b)}

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:

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

- where k is a nonnegative integer,
- $a_0, a_1, \dots, a_k$ 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)
- $(3A0F)_{16} = 3 \cdot 16^3 + 10 \cdot 16^2 + 15 \cdot 16^0 = (14863)_{10}$
-

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 $\neq$ 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

- 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 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 $b = (1011)_2$.

- $a_0 + b_0 = 0 + 1 = 0\cdot2 + 1$, so that $c_0 = 0$ and $s_0 = 1$.
- $a_1 + b_1 + c_0 = 1 + 1 + 0 = 1\cdot2 + 0$, so $c_1 = 1$ and $s_1 = 0$.
- $a_2 + b_2 + c_1 = 1 + 0 + 1 = 1\cdot2 + 0$, so $c_2 = 1$ and $s_2 = 0$.
- $a_3 + b_3 + c_2 = 1 + 1 + 1 = 1\cdot2 + 1$, so $c_3 = 1$ and $s_3 = 1$.
- $s_4 = c_3 = 1$.

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

Addition of Integers

- How do we (humans) add two integers?

- Example:

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

Binary expansions:

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

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)
- c := 0
- for j := 0 to n-1
- begin
-         d := $\lfloor (a_j + b_j + c)/2 \rfloor$
-         s_j := $a_j + b_j + c - 2d$
-         c := d
- end
- s_n := c
- {the binary expansion of the sum is $(s_n s_{n-1} \ldots s_1 s_0)_2$}