

Discrete Structures 2

Lecture 1:

Number Theory

Part 1

Dr Abdullah Alharbi

Second Semester
(1447 AH – 2026 AD)

The lecture content is taken from Discrete Mathematics and Its Applications, 8th edition, by Kenneth H. Rosen. McGraw-Hill, 2019.

Lecture Content:

- **Number Theory**:
  - Divisibility & Modular Arithmetic
  - Integer Representations & Algorithms
  - Primes & Greatest Common Divisors
  - Solving Congruences
  - Applications of Congruences

Number Theory

- Introduction
- Divisibility & Modular Arithmetic:
  - Introduction
  - Division
  - The Division Algorithm
  - Modular Arithmetic
  - Arithmetic Modulo *m*
- Integer Representations & Algorithms:
  - Introduction
  - Representations of Integers:
    - Expansions (Binary, Octal & Hexadecimal)
    - Base Conversion (Binary, Octal & Hexadecimal)

Number Theory: Introduction

- **Number theory** is part of mathematics devoted to the study of the **set of integers** and their **properties**.
- There are several important **applications** of number theory in **CS**:
  - Generate **pseudorandom** numbers.
  - Assign **memory locations** to computer files.
  - Find **check digits** used to detect errors in various kinds of identification numbers.
- **Number theory** has become an essential tool in providing **computer** and **Internet security**.
- **Number theory** plays an essential role both in classical **cryptography** and modern cryptography:
  - Modern cryptography plays an essential role in **electronic communication**:
    - **Cryptographical protocols**: introducing protocols for sharing keys and for sending signed messages.

Divisibility & Modular Arithmetic: Introduction

- **Division** of an **integer** by a **positive integer** produces a **quotient** and a **remainder**.
  - Working with these remainders leads to modular arithmetic.
- **Modular arithmetic** plays an important role in *mathematics* and used throughout *computer science*:
  - Generating **pseudorandom** numbers.
  - Assigning computer **memory locations** to files.
  - Constructing **check digits**.
  - **Encrypting** messages.

Division

- When **one integer** is divided by a **second nonzero integer**, the **quotient** may or may not be an integer.
- **Example**:
  - 12 / 3 = **4** is an integer, whereas 11 / 4 = **2.75** is not.
  - This leads to Definition 1.
- **Definition 1**:

> If a and b are integers with $a \neq 0$, we say that a *divides* b if there is an integer c such that $b = ac$ (or equivalently, if $\frac{b}{a}$ is an integer). When a divides b we say that a is a *factor* or *divisor* of b, and that b is a *multiple* of a. The notation $a \mid b$ denotes that a divides b. We write $a \nmid b$ when a does not divide b.

- ***Remark:*** We can express $a \mid b$ using quantifiers as $\exists c\,(ac = b)$, where the universe of discourse is the **set of integers**

Division

- In **Figure 1**, a number line indicates which integers are divisible by the positive integer d.

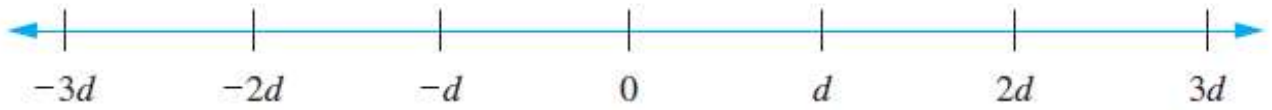


FIGURE 1 **Integers divisible by the positive integer *d*.**

- **Example 1**:

Determine whether **3 | 7** and whether **3 | 12**

Solution: We see that $3 \nmid 7$, because $7/3$ is not an integer. On the other hand, $3 \mid 12$ because $12/3 = 4$.

Division

- **Example 2**:

Let **n** and **d** be positive integers. How many positive integers not exceeding **n** are *divisible* by **d**?

Solution: The positive integers divisible by d are all the integers of the form dk, where k is a positive integer. Hence, the number of positive integers divisible by d that do not exceed n equals the number of integers k with $0 < dk \leq n$, or with $0 < k \leq n/d$. Therefore, there are $\lfloor n/d \rfloor$ positive integers not exceeding n that are divisible by d.

- Some of the basic **properties** of **divisibility** of **integers** are given in **Theorem 1**.

Division

- **THEOREM 1**:

Let a, b, and c be integers, where $a \neq 0$. Then

(*i*) if $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
(*ii*) if $a \mid b$, then $a \mid bc$ for all integers c;
(*iii*) if $a \mid b$ and $b \mid c$, then $a \mid c$.

- **COROLLARY 1**:

If a, b, and c are integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever m and n are integers.

The Division Algorithm

- When an **integer** is **divided** by a **positive integer**, there is a **quotient** and a **remainder**, as the division algorithm shows.

- **THEOREM 2**:

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

- **Remark**: **Theorem 2** is not really an algorithm. (Why not?) Nevertheless, we use its traditional name.

The Division Algorithm

- **Definition 2**:

In the equality given in the division algorithm, d is called the *divisor*, a is called the *dividend*, q is called the *quotient*, and r is called the *remainder*. This notation is used to express the quotient and remainder:

$$q = a \textbf{ div } d, \quad r = a \textbf{ mod } d.$$

- **Remark**: Note that both "a **div** d" and "a **mod** d" for a fixed "**d**" are **functions** on the set of integers.
- Furthermore, when "**a**" is an integer and "**d**" is a positive integer, we have "a **div** d = ⌊a / d⌋" and "a **mod** d = a − d × q"

The Division Algorithm

EXAMPLE 3 What are the quotient and remainder when 101 is divided by 11?

Solution: We have

$$101 = 11 \cdot 9 + 2.$$

Hence, the quotient when 101 is divided by 11 is $9 = 101$ **div** 11, and the remainder is $2 = 101$ **mod** 11.

The Division Algorithm

EXAMPLE 4 What are the quotient and remainder when -11 is divided by 3?

Solution: We have

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

Hence, the quotient when -11 is divided by 3 is $-4 = -11$ **div** 3, and the remainder is $1 = -11$ **mod** 3.

Note that the remainder cannot be negative. Consequently, the remainder is *not* -2, even though

$$-11 = 3(-3) - 2,$$

because $r = -2$ does not satisfy $0 \leq r < 3$.

- **Note** that the integer "**a**" is divisible by the integer "**d**" if and only if the remainder is **zero** when "**a**" is divided by "**d**".

The Division Algorithm

- ***Remark:***

- A programming language may have one, or possibly two, operators for modular arithmetic, denoted by:
  - **mod** (BASIC, Maple, Mathematica, EXCEL, and SQL)
  - **%** (C, C++, Java, and Python)
  - **rem** (Ada and Lisp)
- Be careful when using them, because for "$a < 0$", some of these operators return "$a - m\,[a / m]$" instead of "$a \bmod m = a - m\,\lfloor a / m \rfloor$"
- Also, unlike "$a \bmod m$", some of these operators are defined when "$m < 0$", and even when "$m = 0$".

Modular Arithmetic

- In some situations, we care only about the **remainder** of an **integer** when it is divided by some specified **positive integer**.
- For instance, when we ask what time it will be (on a 24-hour clock) 50 hours from now, we care only about the **remainder** when 50 plus the current hour is divided by 24.
- Because we are often interested only in **remainders**, we have special notations for them.
- We have already introduced the notation "a **mod** m" to represent the **remainder** when an **integer** "**a**" is divided by the **positive integer** "**m**".
- We now introduce a different, but related, notation that indicates that two integers have the same remainder when they are divided by the **positive integer** "**m**".

Modular Arithmetic

- **Definition 3**:

If a and b are integers and m is a positive integer, then 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. We say that $a \equiv b \pmod{m}$ is a **congruence** and that m is its **modulus** (plural **moduli**). If a and b are not congruent modulo m, we write $a \not\equiv b \pmod{m}$.

- Although both notations "**a ≡ b (mod m)**" and "**a mod m = b**" include "**mod**," they represent fundamentally different concepts.
- The **first** represents a relation on the set of integers, whereas the **second** represents a **function**.
- However, the **relation** "**a ≡ b (mod m)**" and the "**mod m**" **function** are closely related, as described in **Theorem 3**.

Modular Arithmetic

- **THEOREM 3**:

Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \bmod m = b \bmod m$.

- Recall that "*a* **mod** *m*" and "*b* **mod** *m*" are the **remainders** when "*a*" and "*b*" are divided by "*m*", respectively.
- Consequently, **Theorem 3** also says that "***a* ≡ *b* (mod *m*)**" if and only if "*a*" and "*b*" have the same remainder when divided by "*m*".

Modular Arithmetic

EXAMPLE 5 Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution: Because 6 divides $17 - 5 = 12$, we see that $17 \equiv 5 \pmod 6$. However, because $24 - 14 = 10$ is not divisible by 6, we see that $24 \not\equiv 14 \pmod 6$.

- The great German mathematician Karl Friedrich **Gauss** developed the concept of **congruences** at the end of the eighteenth century.
- The notion of **congruences** has played an important role in the development of number theory.
- **Theorem 4** provides a useful way to work with **congruences**.

Modular Arithmetic

- **THEOREM 4**:

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 $a = b + km$.

- The set of all integers congruent to an integer "a modulo m" is called the **congruence class** of "a modulo m".

- **Theorem 5** shows that additions and multiplications preserve **congruences**.

Modular Arithmetic

- **THEOREM 5**:

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

$$a + c \equiv b + d \pmod{m} \quad \text{and} \quad ac \equiv bd \pmod{m}.$$

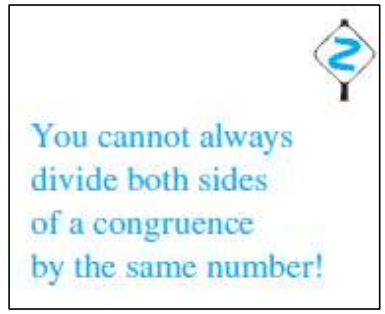
Modular Arithmetic

EXAMPLE 6 Because $7 \equiv 2 \pmod 5$ and $11 \equiv 1 \pmod 5$, it follows from Theorem 5 that

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod 5$$

and that

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod 5.$$



- We must be careful working with **congruences**.
- Some **properties** we may expect to be **true** are **not** *valid*.
- For example, if "**ac ≡ bc (mod m)**", the **congruence** "**a ≡ b (mod m)**" may be **false**.
- Similarly, if "**a ≡ b (mod m)**" and "**c ≡ d (mod m)**", the **congruence** "**ac ≡ bd (mod m)**" may be **false**.

Modular Arithmetic

- **Corollary 2** shows how to find the values of the "**mod** m" function at the **sum** and **product** of two integers using the values of this function at each of these integers.

- **COROLLARY 2:**

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

$$(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$$

and

$$ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m.$$

Modular Arithmetic

EXAMPLE 7 Find the value of $(19^3 \bmod 31)^4 \bmod 23$.

Solution: To compute $(19^3 \bmod 31)^4 \bmod 23$, we will first evaluate $19^3 \bmod 31$. Because $19^3 = 6859$ and $6859 = 221 \cdot 31 + 8$, we have $19^3 \bmod 31 = 6859 \bmod 31 = 8$. So, $(19^3 \bmod 31)^4 \bmod 23 = 8^4 \bmod 23$.

Next, note that $8^4 = 4096$. Because $4096 = 178 \cdot 23 + 2$, we have $4096 \bmod 23 = 2$. Hence, $(19^3 \bmod 31)^4 \bmod 23 = 2$.

Arithmetic Modulo m

We can define arithmetic operations on $\mathbf{Z}_m$, the set of nonnegative integers less than m, that is, the set $\{0, 1, \ldots, m-1\}$. In particular, we define addition of these integers, denoted by $+_m$ by

$$a +_m b = (a + b) \bmod m,$$

where the addition on the right-hand side of this equation is the ordinary addition of integers, and we define multiplication of these integers, denoted by $\cdot_m$ by

$$a \cdot_m b = (a \cdot b) \bmod m,$$

where the multiplication on the right-hand side of this equation is the ordinary multiplication of integers. The operations $+_m$ and $\cdot_m$ are called addition and multiplication modulo m and when we use these operations, we are said to be doing **arithmetic modulo** m.

Arithmetic Modulo m

EXAMPLE 8 Use the definition of addition and multiplication in $\mathbf{Z}_m$ to find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definition of addition modulo 11, we find that

$$7 +_{11} 9 = (7 + 9) \bmod 11 = 16 \bmod 11 = 5,$$

and

$$7 \cdot_{11} 9 = (7 \cdot 9) \bmod 11 = 63 \bmod 11 = 8.$$

Hence, $7 +_{11} 9 = 5$ and $7 \cdot_{11} 9 = 8$.

Arithmetic Modulo m

- The operations "$+_m$" and "$\cdot_m$" satisfy many of the same properties of ordinary addition and multiplication of integers.
- In particular, they satisfy these properties:

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. That is, if a belongs to $\mathbf{Z}_m$, then $a +_m 0 = 0 +_m a = a$ and $a \cdot_m 1 = 1 \cdot_m a = a$.

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

Representations of Integers

- In the modern world, we use ***decimal,*** or ***base* 10**, *notation* to represent **integers**.
- For example, when we write 965, we mean "$9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$"
- We can represent numbers using *any* **base b**, where ***b*** is a **positive integer greater than 1**
- The **bases** "*b* = 2" (**binary**), "*b* = 8" (**octal**) , and "*b* = 16" (**hexadecimal**) are important for **computing** and **communications**
- The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base b Representations

- We can use "**positive integer** *b* **greater than** 1" as a **base**, because of this theorem:
- **Theorem 1**: 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 $a_k \neq 0$

- The a_j, $j = 0, \ldots, k$ are called the **base-*b* digits** of the representation.
- The representation of n given in **Theorem 1** is called the ***base b expansion of n*** and is denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$
- We usually **omit** the **subscript 10** for **base 10 expansions**.

Binary Expansions

- Most computers represent integers and do arithmetic with binary (base 2) expansions of integers.
- In these expansions, the only digits used are 0 and 1
- **Example**: What is the decimal expansion of the integer that has $(1\ 0101\ 1111)_2$ as its binary expansion?
- **Solution**:

$$(1\ 0101\ 1111)_2 = 1\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 351$$

- **Example**: What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?
- **Solution**: $(11011)_2 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 27$

Octal Expansions

- The octal expansion (base 8) uses the digits { 0, 1, 2, 3, 4, 5, 6, 7 }
- **Example**: What is the decimal expansion of the number with octal expansion $(7016)_8$?
- **Solution**: $7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 3598$
- **Example**: What is the decimal expansion of the number with octal expansion $(111)_8$?
- **Solution**: $1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

- The hexadecimal expansion needs 16 digits, but our decimal system provides only 10.
- So letters are used for the additional symbols.
- The hexadecimal system uses the digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }
- The letters A through F represent the decimal numbers 10 through 15
- **Example**: What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?
- **Solution**: $2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 175627$
- **Example**: What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?
- **Solution**: $14 \cdot 16^1 + 5 \cdot 16^0 = 224 + 5 = 229$

Base Conversion

- To construct the **base** *b* **expansion of an integer** *n*:
  - Divide *n* by *b* to obtain a quotient and remainder "$n = b\ q_0 + a_0$" $0 \le a_0 \le b$
  - The remainder, a_0, is the **rightmost digit** in the base *b* expansion of *n*
  - Next, **divide** q_0 by *b* "$q_0 = b\ q_1 + a_1$" $0 \le a_1 \le b$
  - The remainder, a_1, is the **second digit from the right** in the base *b* expansion of *n*
  - Continue by successively dividing the quotients by *b*, obtaining the additional base *b* digits as the remainder.
  - The **process terminates** when the quotient is 0

Base Conversion

- **Example**: Find the octal expansion of $(12345)_{10}$
- **Solution**: Successively dividing by 8 gives:
  - $12345 = 8 \cdot 1543 + 1$
  - $1543 = 8 \cdot 192 + 7$
  - $192 = 8 \cdot 24 + 0$
  - $24 = 8 \cdot 3 + 0$
  - $3 = 8 \cdot 0 + 3$
- The remainders are the digits from right to left yielding $(30071)_8$

$$(12345)_{10} = (30071)_8$$

EXAMPLE Find the hexadecimal expansion of $(177130)_{10}$.

Solution: First divide 177130 by 16 to obtain

$$177130 = 16 \cdot 11070 + 10.$$

Successively dividing quotients by 16 gives

$$\begin{aligned} 11070 &= 16 \cdot 691 + 14, \\ 691 &= 16 \cdot 43 + 3, \\ 43 &= 16 \cdot 2 + 11, \\ 2 &= 16 \cdot 0 + 2. \end{aligned}$$

The successive remainders that we have found, 10, 14, 3, 11, 2, give us the digits from the right to the left of 177130 in the hexadecimal (base 16) expansion of $(177130)_{10}$. It follows that

$$(177130)_{10} = (2B3EA)_{16}.$$

(Recall that the integers 10, 11, and 14 correspond to the hexadecimal digits A, B, and E, respectively.)

EXAMPLE Find the binary expansion of $(241)_{10}$.

Solution: First divide 241 by 2 to obtain

$$241 = 2 \cdot 120 + 1.$$

Successively dividing quotients by 2 gives

$$\begin{aligned} 120 &= 2 \cdot 60 + 0, \\ 60 &= 2 \cdot 30 + 0, \\ 30 &= 2 \cdot 15 + 0, \\ 15 &= 2 \cdot 7 + 1, \\ 7 &= 2 \cdot 3 + 1, \\ 3 &= 2 \cdot 1 + 1, \\ 1 &= 2 \cdot 0 + 1. \end{aligned}$$

The successive remainders that we have found, 1, 0, 0, 0, 1, 1, 1, 1, are the digits from the right to the left in the binary (base 2) expansion of $(241)_{10}$. Hence,

$$(241)_{10} = (1111\ 0001)_2.$$

The pseudocode given in Algorithm 1 finds the base b expansion $(a_{k-1} \ldots a_1 a_0)_b$ of the integer n.

ALGORITHM 1 **Constructing Base b Expansions**

procedure *base b expansion*(n, b: positive integers with $b > 1$)
$q := n$
$k := 0$
while ($q \neq 0$)
$a_k := q$ **mod** b
$q := q$ **div** b
$k := k + 1$
return($a_{k-1}, \ldots, a_1, a_0$) {$(a_{k-1} \ldots a_1 a_0)_b$ is base b expansion of n}

- q represents the **quotient** obtained by successive divisions by b, starting with $q = n$
- The digits in the base b expansion are the remainders of the division given by "q **mod** b"
- The algorithm terminates when "$q = 0$" is reached

Comparison of **Hexadecimal**, **Octal**, and **Binary** Representations

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.																
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Initial 0s are not shown

- Each octal digit corresponds to a block of 3 binary digits.
- Each hexadecimal digit corresponds to a block of 4 binary digits.
- So, conversion between binary, octal, and hexadecimal is easy.

Conversion Between **Binary**, **Octal**, and **Hexadecimal** Expansions

- **Example**:

  Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$

- **Solution**:

  - To convert to octal, we **group the digits** into blocks of **three** $(011\ 111\ 010\ 111\ 100)_2$, **adding initial 0s as needed**.
  - The **blocks** from left to right correspond to the digits 3, 7, 2, 7, and 4. Hence, the solution is $(37274)_8$
  - To convert to hexadecimal, we **group the digits** into blocks of **four** $(0011\ 1110\ 1011\ 1100)_2$, **adding initial 0s as needed**.
  - The **blocks** from left to right correspond to the digits 3, E, B, and C. Hence, the solution is $(3EBC)_{16}$

Any Questions?

Have a look at the "Extra Examples" on the textbook's website.

Keep reviewing this lecture *regularly, as well as other lectures*.

Have a look at the *exercises* at the end of the sections we covered in this lecture in Chapter 4 and do your best with those ones similar to what we presented in this lecture.

- You should get more *examples* and *exercises* in the *tutorials*.

Ask *me or ask your peers* if you have any questions.

- Please, do not wait until the end of the semester or even right before exams.

All the best!