數論 Number Theory

第四章

Outline

- 整除性和模術性 (Divisibility & Modular Arithmetic)
- 整數表示及算法 (Integer Representations and Algorithms)
- 素數和最大公約數 (Primes and Greatest Common Divisors)

整除性 Divisibility (4.1)

$$\forall \boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{Z}(\boldsymbol{a}|\boldsymbol{b} \leftrightarrow \exists \boldsymbol{c}(\boldsymbol{ac} = \boldsymbol{b}))$$

例如:

- $3 \mid 15$
  - "3整除15"
  - "3 divides 15" or "15 is divisible by 3"
- $3 \nmid 22$
  - "3不整除22"
  - "3 does not divide 15" or "15 is not divisible by 3"

+ 整除性性質 Properties of Divisibility

- (4.1定理1:)
  已知 $a, b, c \in \mathbf{Z}$, 其中 $a \neq 0$, 那麼:
  i. $(a|b \wedge a|c) \rightarrow a \mid (b+c)$;
  ii. $a|b \rightarrow \forall c(a|bc)$;
  iii. $(a|b \wedge b|c) \rightarrow a|c$.

+ 除法算法 Division Algorithm

- **除法定理**：若 $a \in \mathbf{Z}, d \in \mathbf{Z}^+$, 那麼存在 $q, r \in \mathbf{Z}$ 滿足：

$$a = dq + r\ ,$$

其中 $0 \leq \boldsymbol{r} < \boldsymbol{d}$

dividend
(被除數)

divisor
(除數)

quotient
(商)

remainder
(余數)

函數 **div** 與 **mod** 的定義:

$q = a \ \mathbf{div}\ d$

$r = a \ \mathbf{mod}\ d$

對應: $\frac{a}{d} = q + \frac{r}{d}$

例1

函數 **div** 與 **mod** 的定義:

$q = a \textbf{ div } d$

$r = a \textbf{ mod } d$

1. 以 div 及 mod 表示並求出商與余數:
   - A. 101除以11 $101 \text{ div } 11 = 9$ $101 \text{ mod } 11 = 2$
   - B. -11除以3 $-11 \text{ div } 3 = -3$ $-11 \text{ mod } 3 = -2$

2. 求下列各值:
   - A. 130 div 3 43
   - B. 130 mod 3 1
   - C. -111 div 9 -12
   - D. -111 mod 9 6

同余式 Congruence

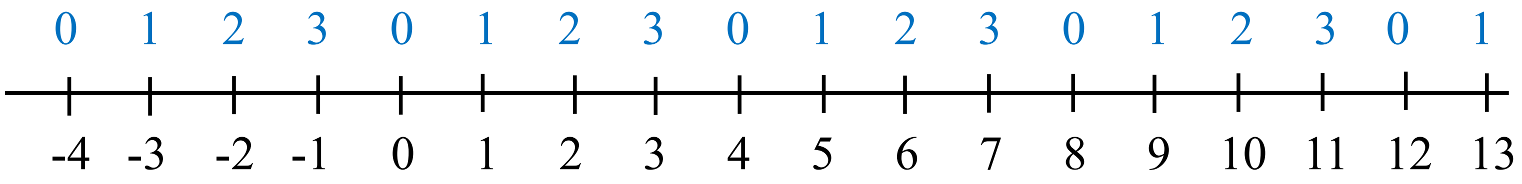
若 $a, b \in \mathbf{Z}, m \in \mathbf{Z}^+$, 那麼:

- 若 $m|a-b$，則 “a is congruent to b modulo m (a 模 m 同余 b)”

  記作: $a \equiv b \pmod{m}$.

- 稱 $a \equiv b \pmod{m}$ 為同余式(congruence), 其中 m 為它的模(modulus).

- 例如:
  - $13 \equiv 5 \pmod{4}$;
  - $12 \not\equiv 14 \pmod{4}$.



同余式的定理及性質(4.1)

- $\forall a, b, c, d, k \in \mathbf{Z}, m \in \mathbf{Z}^+$:
  - $a \equiv b \pmod{m} \leftrightarrow (a \bmod m = b \bmod m)$ (4.1定理3)
  - $a \equiv b \pmod{m} \leftrightarrow \exists k(a = b + km)$ (4.1定理4)
  - (4.1定理5)

$$a \equiv b \pmod{m} \quad and \quad c \equiv d \pmod{m}$$

$$IFF$$

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

模m算術 (Arithmetic Modulo m)

- U: $\{0, 1, 2, \ldots, m\}$
- $+_m$ 加法運算:

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

- $\cdot_m$ 乘法運算:

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

例2

- 求:

A. $4 +_3 5$ 　 $9 \bmod 3 = 0$

B. $4 \cdot_3 5$ 　 $20 \bmod 3 = 2$

+ 整數表示與算法 (4.2)

- 十進制展開式 Decimal expansion
- 二進制展開式 Binary expansion
- 八進制展開式 Octal expansion
- 十六進制展開式 Hexadecimal expansion

進制的展開式 Expansion

- n 的 b 進制展開式 (Base b Expansion of n):
  令 $b \in \mathbf{Z}, b > 1$, $k \in \boldsymbol{N}$, 則對於正整數 n 有:
  $$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

- 例如:
  - 315的十進制展開式為:
    $$315 = 3 \times 10^2 + 1 \times 10^1 + 5 = (315)_{10}$$
  - 315的八進制展開式為:
    $$315 = 4 \times 8^2 + 7 \times 8^1 + 3 = (473)_8$$

例3: 其他進制轉換到十進制

A. 求 $(1010\ 1001)_2$ 的十進制展開式。

B. 求 $(4103)_8$ 的十進制展開式。

C. 求 $(2B0C)_{16}$ 的十進制展開式。

step 1 Convert $(1010\ 1001)_2$ to decimal by multiplying each digit by 2 raised to the power of its position, starting from the right with position 0

step 2 $(1010\ 1001)_2 = 1\cdot 2^7 + 0\cdot 2^6 + 1\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$

step 3 Calculate the powers of 2: $1\cdot 128 + 0\cdot 64 + 1\cdot 32 + 0\cdot 16 + 1\cdot 8 + 0\cdot 4 + 0\cdot 2 + 1\cdot 1$

step 4 Add the results: $128 + 0 + 32 + 0 + 8 + 0 + 0 + 1 = 169$

step 1 Convert $(4103)_8$ to decimal by multiplying each digit by 8 raised to the power of its position, starting from the right with position 0

step 2 $(4103)_8 = 4\cdot 8^3 + 1\cdot 8^2 + 0\cdot 8^1 + 3\cdot 8^0$

step 3 Calculate the powers of 8: $4\cdot 512 + 1\cdot 64 + 0\cdot 8 + 3\cdot 1$

step 4 Add the results: $2048 + 64 + 0 + 3 = 2115$

step 1 Convert each hexadecimal digit to its decimal equivalent. The hexadecimal number is $(2B0C)_{16}$

step 2 Starting from the right, each digit represents increasing powers of 16. The rightmost digit is 16^0, the next is 16^1, and so on

step 3 Calculate the decimal value for each digit: 2×16^3, $B\times 16^2$, 0×16^1, $C\times 16^0$. Note that B in hexadecimal is 11 in decimal, and C is 12

step 4 Perform the calculations: $2\times 16^3 = 2\times 4096$, $11\times 16^2 = 11\times 256$, $0\times 16^1 = 0$, $12\times 16^0 = 12$

step 5 Add the results of each calculation to get the final decimal value: $8192 + 2816 + 0 + 12$

step 6 Sum the values: $8192 + 2816 + 12 = 11020$

+ 範例: 從十進制轉換到其他進制

求$(62)_{10}$的二進制展開式。

$$62 = 2 \cdot 31 + \underline{0}$$
$$31 = 2 \cdot 15 + \underline{1}$$
$$15 = 2 \cdot 7 + \underline{1}$$
$$7 = 2 \cdot 3 + \underline{1}$$
$$3 = 2 \cdot 1 + \underline{1}$$
$$1 = 2 \cdot \boxed{0} + \underline{1}$$

直到得到0為止

$$\therefore (62)_{10} = (11\,1110)_2$$

例4

A. 求 $(6234)_{10}$ 的八進制展開式。 14132 使用短除法

B. 求 $(6234)_{10}$ 的十六進制展開式。 185A

step 1 Convert the decimal number 6234 to hexadecimal

step 2 Divide 6234 by 16 and find the remainder

step 3 The quotient is 389 with a remainder of 10, which is 'A' in hexadecimal

step 4 Repeat the process by dividing the quotient by 16 until the quotient is 0

step 5 The next quotient is 24 with a remainder of 5, which is '5' in hexadecimal

step 6 The final quotient is 1 with a remainder of 8, which is '8' in hexadecimal

step 7 Write down the remainders in reverse order to get the hexadecimal number

+ 不同進制之間的轉換

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

- 八進制和十六進制之間的轉換可先轉換成二進制更容易。

例5: 不同進制之間的轉換

A. 求$(11\ 1110\ 1011)_2$的八進制展開式。(1753)

B. 求$(6234)_8$的二進制展開式。(1100 1001 1100)

C. 求$(6234)_8$的十六進制展開式。(C9C)

生成不同進制展開式的算法

- 偽代碼(Pseudocode):

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}

不同進制加法運算的算法

- 偽代碼(Pseudocode):

procedure $add(a, b$: positive integers)
{the binary expansions of a and b are $(a_{n-1}, a_{n-2}, \ldots, a_0)_2$ and $(b_{n-1}, b_{n-2}, \ldots, b_0)_2$, respectively}
$c := 0$
for $j := 0$ to $n - 1$
    $d := \lfloor (a_j + b_j + c)/2 \rfloor$
    $s_j := a_j + b_j + c - 2d$
    $c := d$
$s_n := c$
return$(s_0, s_1, \ldots, s_n)$ {the binary expansion of the sum is $(s_n, s_{n-1}, \ldots, s_0)_2$}

- 算法複雜度: $f(n)$ is $O(n)$.

不同進制乘法運算的算法

- 偽代碼(Pseudocode):

procedure *multiply*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1},a_{n-2},\ldots,a_0)_2$ and $(b_{n-1},b_{n-2},\ldots,b_0)_2$, respectively}
for $j := 0$ to $n-1$
    if $b_j = 1$ **then** $c_j = a$ shifted j places
    else $c_j := 0$
{$c_o, c_1, \ldots, c_{n-1}$ are the partial products}
$p := 0$
for $j := 0$ to $n-1$
  $p := p + c_j$
return p {p is the value of ab}

- 算法複雜度: $f(n)$ is $O(n^2)$.

素數和最大公約數(4.3)

- 大於 1 且只能被 1 及 p 整除的正整數 p 稱為素數(Prime)。
- 大於 1 且不是素數的正整數為合數(Composite)。

Prime

Composite

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60

+ 算術基本定理 The Fundamental Theorem of Arithmetic

- 每個大於 1 的整數都能唯一的寫成兩個或以上非遞減序列素數的乘積。

  其表示稱作該數的素因子分解式(prime factorization)

- 例如：
  - $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
  - $641 = 641$
  - $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
  - $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

+ 最大公約數 Greatest Common Divisor

- 能整除兩個整數的最大整數稱為這兩個整數的最大公約數(greatest common divisor).
  - 即：令 a 和 b 為兩個整數, 不全為 0. 能使 $d \mid a$ 及 $d \mid b$ 的最大整數 d 為 a 和 b 的最大公約數, 記作 gcd(a, b).
    - 例如： gcd(24, 36) = 12； gcd(17, 22) = 1

- 若 gcd(a, b) = 1, 稱 a, b 是互素的 (relatively prime).
    - 例如： gcd(17, 22) = 1 ； gcd(10, 21) = 1

+ 最小公倍數 Least Common Multiple

- 能被兩個正整數整除的最小正整數稱為這兩個整數的最小公倍數(Least Common Multiple).
  - 即：令 a 和 b 為兩個正整數. 能使 $a \mid m$ 及 $b \mid m$ 的最小正整數 m 為 a 和 b 的最小公倍數, 記作 lcm(a, b).
    - 例如：lcm(12,24) = 24;　　　　lcm(9,24) = 72

+ 利用素因子分解式求兩數的 gcd 與 lcm

- 已知兩數a,b的素因子分解式分別為:

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}, \qquad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

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

+ 例6

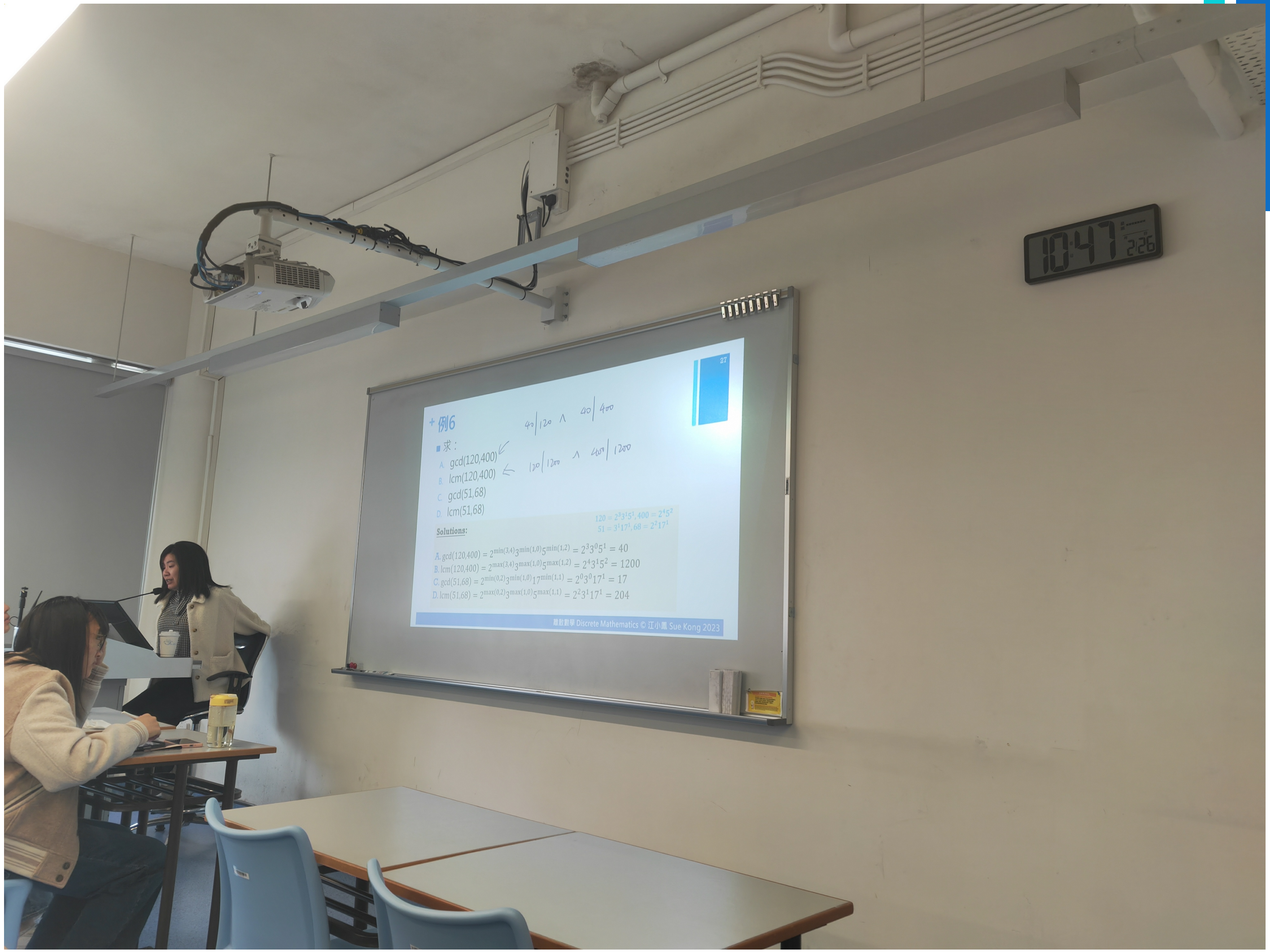
- 求：

A. gcd(120,400)

B. lcm(120,400)

C. gcd(51,68)

D. lcm(51,68)



歐幾里得算法 Euclidean Algorithm

- 偽代碼：

推論 1: 若 $a = bq + r$, 其中 $a, b, q,$ 及 r 皆為整數. 則 $\gcd(a,b) = \gcd(b,r)$.

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

例如：

$\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$

$287 = 91 \cdot 3 + 14$
$91 = 14 \cdot 6 + 7$
$14 = 7 \cdot 2 + 0$

例7

- 利用歐幾里得算法(Euclidean Algorithm)求：
  A. gcd(12345, 67890) = 15
  B. gcd(54321, 9876) = 3

+ 教材對應閱讀章節及練習

- 4.1-4.2(~Example7), 4.3
- 對應習題: (可視個人情況定量)
  - 4.1: 1,9-10, 13-29
  - 4.2: 1-12
  - 4.3: 1-4, 14-17, 24-27, 32-35