Advanced Competitive Programming

國立成功大學ACM-ICPC程式競賽培訓隊
nckuacm@imslab.org

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

Number Theory

Outline

- Modular arithmetic
- Greatest Common Divisor

Outline

- Modular arithmetic
- Greatest Common Divisor

餘數

對於整數 a
除以整數 b

除不盡的部分就是餘數

餘數

對於整數 a
除以整數 b

除不盡的部分就是餘數

例如 6 / 4 = 1 ... 2
6 = 1 * 4 + 2

同餘

某些整數除以 n
他們的餘數相同 就稱為**同餘**

同餘

某些整數除以 n
他們的餘數相同 就稱為**同餘 n**

例如 6 與 10 除以 4 都餘 2

同餘

某些整數除以 n
他們的餘數相同 就稱為**同餘 n**

a 與 b 同餘 n
記為 $a \equiv b \bmod n$

同餘

某些整數除以 n
他們的餘數相同 就稱為**同餘 n**

6 與 10 同餘 4
記為 $6 \equiv 10 \bmod 4$

同餘運算

對於

- $a \equiv b \bmod n$
- $c \equiv d \bmod n$

同餘運算

對於

- $a \equiv b \bmod n$
- $c \equiv d \bmod n$

有

$a+c \equiv b+d \bmod n$

同餘運算

對於

- $a \equiv b \bmod n$
- $c \equiv d \bmod n$

有

$a+c \equiv b+d \bmod n$

$a \times c \equiv b \times d \bmod n$

同餘運算

例如

- $6 \equiv 10 \mod 4$
- $5 \equiv 11 \mod 4$

有

$6+5 \equiv 10+11 \mod 4$

同餘運算

例如

- $6 \equiv 10 \bmod 4$
- $5 \equiv 11 \bmod 4$

有

$6+5 \equiv 10+11 \bmod 4$

$6\times 5 \equiv 10\times 11 \bmod 4$

歐拉定理

a, n 互質 則

$$a^{\phi(n)} \equiv 1 \bmod n$$

$\phi(n)$ 表示與 n 互質且小於 n 的正整數的個數
例如 1, 5 和 6 互質， $\phi(6)=2$

費馬小定理

a, p 互質，且 p 為質數 則

$$a^{p-1} \equiv 1 \bmod p$$

Outline

- Modular arithmetic
- Greatest Common Divisor

最大公因數

整數 a, b，它倆各自有自己的因數

取**相同的因數**中**最大**的數，即 gcd(a, b)

性質

gcd(a, b) = gcd(b, a)

性質

$\gcd(a, b) = \gcd(b, a)$

$\gcd(a, 0) = |a|$

性質

$$\gcd(a, b) = \gcd(b, a)$$

$$\gcd(a, 0) = |a|$$

$$\gcd(a, b) = \gcd(a, b\%a)$$

性質

$\gcd(a, b) = \gcd(b, a)$

$\gcd(a, 0) = |a|$

$\gcd(a, b) = \gcd(a, b\%a)$

其中 b%a 表示 b 除以 a 的**餘數**

輾轉相除法

找出 gcd(a, b) 的值 (normal form)

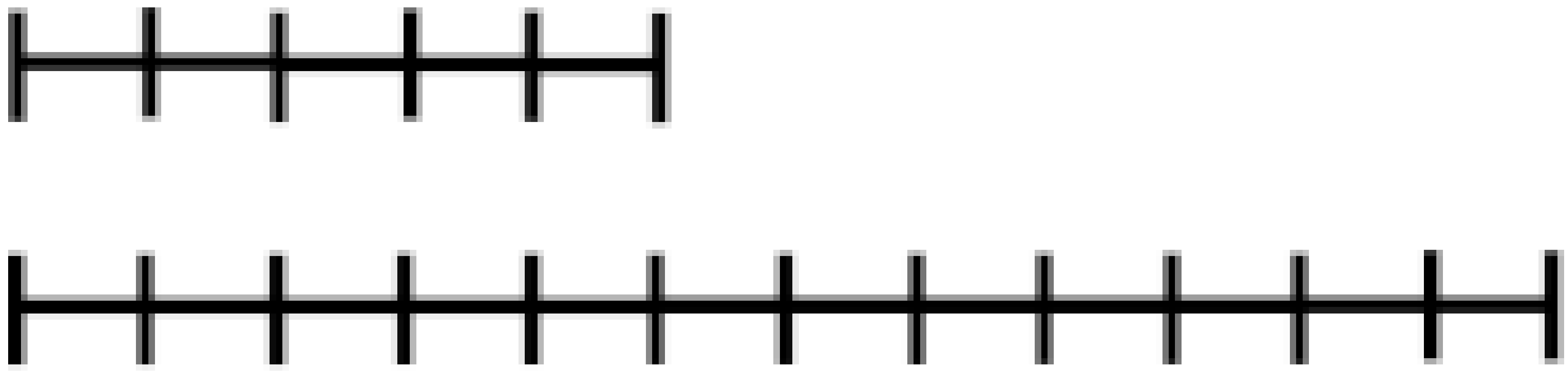
輾轉相除法

找出 gcd(a, b) 的值 (normal form)

根據 GCD 的性質，有

gcd(a, b) = g(b, a) = g(b%a, a)

輾轉相除法



輾轉相除法

找出 gcd(a, b) 的值 (normal form)

根據 GCD 的性質，有

gcd(a, b) = g(b, a) = g(b%a, a)

輾轉相除法

找出 gcd(a, b) 的值 (normal form)

根據 GCD 的性質，有

gcd(a, b) = g(b, a) = g(b%a, a)

其中 b%a 只有當 b ≥ a 時才有變化

輾轉相除法

$\gcd(a_0, b_0) = \gcd(b_0\%a_0, a_0)$, 設 $b_1 = b_0\%a_0$

$\gcd(b_1, a_0) = \gcd(a_0\%b_1, b_1)$, 設 $a_1 = a_0\%b_1$

$\gcd(a_1, b_1) = \gcd(b_1\%a_1, a_1)$, 設 $b_2 = b_1\%a_1$

$\vdots$

$\gcd(\mathbf{0}, b_n) = |b_n|$

輾轉相除法 (例)

gcd(15, 42) = gcd(42%15, 15), 12 = 42%15
gcd(12, 15) = gcd(15%12, 12), 03 = 15%12
gcd(03, 12) = gcd(12%03, 03), 00 = 12%03

gcd(**0**, 3) = 3

輾轉相除法 (實作)

```
int gcd(int a, int b) {
  return a? gcd(b%a, a) : b;
}
```

貝祖定理

對於所有整數 a, b，

存在整數 x, y 使得 $ax+by = \gcd(a, b)$

擴展歐幾里得演算法

找到 gcd(a, b) 同時，找出 x, y

擴展歐幾里得演算法

找到 gcd(a, b) 同時，找出 x, y

為了簡潔，令 $g = \gcd(a, b)$

擴展歐幾里得演算法

考慮輾轉相除法

找到 g 的時候

擴展歐幾里得演算法

考慮輾轉相除法

找到 g 的時候，根據貝祖定理

$$\mathbf{0} \cdot x_n + \mathbf{g} \cdot y_n = g$$

擴展歐幾里得演算法

考慮輾轉相除法

找到 g 的時候，根據貝祖定理

$$\mathbf{0}\cdot x + \mathbf{g}\cdot y = g$$

明顯有 x 為任意整數， y 為 1
(別搞混符號，這裡 x, y 不是原問題要求的!!!)

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有
(這裡 a, b 不是原問題的!! 而是過程中的)

$$g = \gcd(a, b) = \gcd(b\%a, a)$$

(根據 GCD 性質)

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = \gcd(a, b) = \gcd(b\%a, a)$$

$$\Rightarrow \quad g = (b\%a)\,x + a\,y$$

(貝祖定理)

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = (b\%a)\,x + a\,y$$

$$\Rightarrow \quad g = (b - \lfloor b/a \rfloor \cdot a)\,x + a\,y$$

(b%a 的定義)

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = (b - \lfloor b/a \rfloor \cdot a)\, x + a\, y$$

$$\Rightarrow \quad g = b\, x + a\, (y - \lfloor b/a \rfloor \cdot x)$$

(移個項)

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = b\,x + a\,(y - \lfloor b/a \rfloor \cdot x)$$

對於 a 與 b 的貝祖等式

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = b\,x + a\,(y - \lfloor b/a \rfloor \cdot x)$$

對於 a 與 b 的貝祖等式

$x_t = (y - \lfloor b/a \rfloor \cdot x)$

$y_t = x$

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = b\,x + a\,(y - \lfloor b/a \rfloor \cdot x)$$

對於 a 與 b 的貝祖等式 $a\,x_t + b\,y_t = g$

$x_t = (y - \lfloor b/a \rfloor \cdot x)$

$y_t = x$

擴展歐幾里得演算法(實作)

```
int gcd(int a, int b, int &x, int &y) {
   if(!a) {
      x = 0, y = 1; // x 為任意整數
      return b;
   }

   int g = gcd(b%a, a, x, y);

   int xp = y - b/a * x, yp = x;
   x = xp, y = yp;

   return g;
}
```

擴展歐幾里得演算法(實作)

```
int gcd(int a, int b, int &x, int &y) {
  if(!a) {
    x = 0, y = 1; // x 為任意整數
    return b;
  }
  int g = gcd(b%a, a, x, y);
  int xp = y - b/a * x, yp = x;
  x = xp, y = yp;
  return g;
}
```

擴展歐幾里得演算法

考慮輾轉相除法

找到 g 的時候，根據貝祖定理

$$\mathbf{0} \cdot x + \mathbf{g} \cdot y = g$$

明顯有 x 為任意整數， y 為 1
(別搞混符號，這裡 x, y 不是原問題要求的!!!)

擴展歐幾里得演算法(實作)

```
int gcd(int a, int b, int &x, int &y) {
   if(!a) {
      x = 0, y = 1; // x 為任意整數
      return b;
   }

   int g = gcd(b%a, a, x, y);

   int xp = y - b/a * x, yp = x;
   x = xp, y = yp;

   return g;
}
```

擴展歐幾里得演算法

考慮輾轉相除法，對**過程中**任意 a, b 有

$$g = b\,x + a\,(y - \lfloor b/a \rfloor \cdot x)$$

對於 a 與 b 的貝祖等式 $a\,x_t + b\,y_t = g$

$x_t = (y - \lfloor b/a \rfloor \cdot x)$

$y_t = x$

Questions?