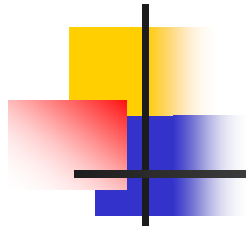




# Chapter 10

## Applications of Primitive Roots and the Order of an Integer

邱錫彥 老師



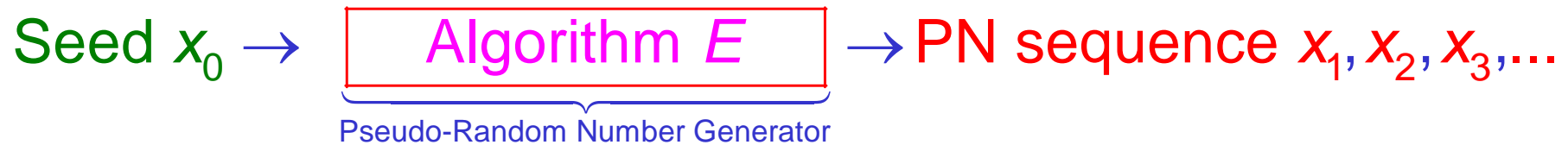
# Contents

---

10.1 Pseudo-random numbers.....3



# 10.1 Pseudo-random Numbers



## Conditions:

1. Long period.
2. Good statistical properties.
3. Unpredictable.
4. Large Linear complexity and good linear complexity profile.



# Method to generate PN sequence

## 1. Middle-square Method (by J.V. Neumann)

Let  $f$  be a truncated function such that

the output  $f(a)$  is the middle digits of  $a$  and  $|f(a)| = |a|/2$

Ex: Let  $a = 37687321$ , then  $f(a) = 6873$ .



# 1. Middle-square Method

Algorithm:  $x_0$  is the seed

$$x_1 = f(x_0^2)$$

$$x_2 = f(x_1^2)$$

⋮

$$x_{i+1} = f(x_i^2)$$

EX :  $x_0 = 6139$ ,  $x_1 = f(37687321) = 6873$ ,

$x_2 = f(47238129) = 2381$ ,

$x_3 = f(5669161) = 6691$

⋮



Remark :

1. When  $x_0$  is know, then the **entire sequence is determined.**
2. The sequence appears to be **random** and is useful for **computer simulation.**
3. How to choose  $x_0$  such that the **period** is as long as possible ?

Ex:

If  $x_0 = 4100$ , then

$$x_1 = 8100, x_2 = 6100, x_3 = 2100, x_4 = 4100.$$

$\Rightarrow$  period = 4



## 2. Linear congruential method (LCM)

### Linear congruential method (LCM)

seed:  $x_0$

parameters of algorithm :

$m$ : modulus

$a$ : multiplier ,  $2 \leq a < m$

$c$ : increment .

algorithm :

$$x_{n+1} = ax_n + c \text{ mod } m, \text{ for } n = 0, 1, 2, \dots$$



# Linear congruential method (LCM)

Ex :

Let  $m = 9$ ,  $a = 7$ ,  $c = 4$  and  $x_0 = 3$ , then the output sequence is

$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \dots$

period = 9 (=  $m$ )

Ex :

let  $m = 12$ ,  $a = 3$ ,  $c = 4$  and  $x_0 = 5$ , then

$x_1 = 7$ ,  $x_2 = 1$ ,  $x_3 = 7$ ,  $x_4 = 1$ , ...

period = 2





# Linear congruential method (LCM)

Remark :

1. If we want to generate PN between 0 and 1, then we can make the output be  $x_i/m$ ,  
 $i = 1, 2, \dots$
2. Assume the probability density function (pdf) of the PN sequence with another type of pdf. Then we first generate a PN with uniform distribution and then transform it to the type of distribution that we want.

A decorative graphic on the left side of the slide, featuring overlapping yellow, red, and blue squares with a black crosshair.

## Linear congruential method (LCM)

---

3. How to choose the parameters  $m$ ,  $a$  and  $c$  such that
  - a. The computational speed is as fast as possible.
  - b. The period is as long as possible ( $= m$ )
4. In order to make the computational speed is as fast as possible, we usually make  $m = 2^n$  or  $2^{n-1}$ , where  $n$  is the bit length of a word in computer.



# Linear congruential method (LCM)

Thm :

the sequence generate by LCM is given by

$$x_k = a^k x_0 + c \frac{a^k - 1}{a - 1} \pmod{m}$$

Proof :

By mathematical induction, when  $k = 1$ , it is true.

Assume that it is true for the  $k$ th term , i.e..

$$x_k = a^k x_0 + c \frac{a^k - 1}{a - 1} \pmod{m}$$

$$\Rightarrow x_{k+1} = ax_k + c \pmod{m}$$



# Linear congruential method (LCM)

$$\begin{aligned}\Rightarrow x_{k+1} &= a \left[ a^k x_0 + c \frac{a^k - 1}{a - 1} \right] + c \pmod{m} \\ &= a^{k+1} x_0 + c \frac{a(a^k - 1) + (a - 1)}{a - 1} \pmod{m} \\ &= a^{k+1} x_0 + c \frac{a^{k+1} - 1}{a - 1} \pmod{m}\end{aligned}$$

∴ It is also true for  $(k+1)$ th term.

Thus, the formula is correct for all  $k$ .



# Linear congruential method (LCM)

Thm :

The PN seunce generated by LCM has period length  $m$  iff.

(1)  $(c, m) = 1$

(2)  $a = 1 \pmod p$ , for all primes dividing  $m$

(3)  $a = 1 \pmod 4$  if  $4|m$

Generalization: (**kth order LCM**)

Seed:  $x_0, x_1, x_2, \dots, x_{k-1}$

Parameters:  $a_0, a_1, a_2, \dots, a_{k-1}$ ;  $c$  and  $m$ .

Algorithm:  $x_j = a_0 x_{j-k} + a_1 x_{j-k+1} + a_2 x_{j-k+2} + \dots + a_{k-1} x_{j-1} + c \pmod m,$

$$j \geq k$$



## 3. Simplified LCM

(Pure multiplicative Congruential method)

### Simplified LCM (Pure multiplicative Congruential method)

Seed:  $x_0$

Parameters:  $a$  and  $m$

Algorithm :

$$x_{n+1} = ax_n \bmod m; n \geq 1 \text{ or } x_n = a^n x_0 \bmod m.$$

Let  $l$  be the period length of pure multiplicative generator.  
Then  $l$  is the smallest positive integer such that

$$x_0 = a^l x_0 \bmod m.$$

If  $(x_0, m) = 1$ , then

$$a^l = 1 \bmod m.$$

Thus, the largest possible period length is  $\lambda(m)$ ,  
where  $\lambda(m)$  is the minimum universal exponent modulo  $m$ .



# Simplified LCM

Remark:

- In order to make  $l$  be as large as possible, we should choose  $a$  and  $m$  such that
$$\text{ord}_m a = \lambda(m).$$
- If  $m$  is prime, then  $a$  should be a primitive root modulo  $m$ .
- $\lambda(m)$  should not be too small.

Eg. The integer 7 is a primitive root of  $M_{31} = 2^{31} - 1$ .



# Security of PN sequences

- From the viewpoint of **security**, all the PN sequences generated by LCM are **not secure enough** even if the parameters in LCM are unknown.
- A **more secure method** is to use a **truncated function** to the output PN sequence , i.e. The output is  $f(x_i)$  instead of  $x_i$  , where  $f$  is a truncated function.