บทที่ 2
การสร้างตัวเลขสุ่ม

การสร้างตัวเลข เราสามารถทำได้หลายแบบหลายวิธี [1] และตัวเลขที่สร้างขึ้นมาได้ มีลักษณะต่าง ๆ กันออกไป ซึ่งขึ้นอยู่กับวิธีที่ใช้ในการสร้าง ดังนั้นก่อนที่เราจะสร้างตัวเลข เราจึงควรจะทราบถึงลักษณะของตัวเลขที่เราต้องการเสียก่อน เช่นเราต้องการตัวเลขสุ่ม (Random Number) การสร้างตัวเลขสุ่ม เราทำได้โดยอาศัยทฤษฎีทางคณิตศาสตร์เข้าช่วยในการสร้างได้หลายวิธี ตัวเลขที่สร้างขึ้นมาจะต้องมี ค่าต่าง ๆ กัน ตัวเลขแต่ละตัวจะต้องมีโอกาสเกิดขึ้นเท่า ๆ กัน (ความน่าจะเป็นของแต่ละค่าที่จะเกิดขึ้นคือ $\frac{1}{n}$) และค่าใหม่ที่จะเกิดขึ้นต้องเป็นอิสระจากค่าที่เกิดขึ้นก่อน ในทางสถิติยอมหมายความว่า ตัวเลขจะมีการกระจายแบบ Uniform และ independent ซึ่งเราถือว่าตัวเลขที่มีลักษณะตามที่กล่าวแล้วจะเป็นตัวเลขสุ่ม

วิธีสร้างตัวเลขสุ่มเราทำได้หลายวิธี แต่วิธีที่เป็นที่รู้จักดี คือ

1. วิธี Middle - square
2. วิธี Power Residue
3. วิธี Linear Congruence
   - 3.1 วิธี Multiplicative
   - 3.2 วิธี Mixed

2.1 วิธี Middle - square [2]

ในปี ค.ศ. 1946 John Von Neuman ได้เสนอวิธีการที่เรียกว่า Middle - square ขึ้นมาเป็นครั้งแรก หลักการของวิธีนี้มีอยู่ว่ากำหนดค่าเริ่มต้น (initial number) X_0 ซึ่งประกอบด้วยตัวเลขที่เป็นคู่ เรากำหนดสัญญลักษณ์ให้เป็น p - digit ตัวเลขตัวต่อไปของชุดลำดับคือ X_1 ได้จากการเอา X_0 มายกกำลัง 2 แล้วเราจะได้ตัวเลข 2 p - digit เรา

---

1. Jame R. Emshoff and Roger L. Sission; Design and Use of Computer Simulation Models, page 170 - 173
2. Joe H. Mize and J. Grady Cox, Essentials of Simulation page 63-64

นำเอาตัวกลางของ X_0 มาเป็น X_1 วิธีนี้จะดำเนินต่อไปเรื่อย ๆ ไป เราจะได้ชุดลำดับของตัวเลขขึ้นมาชุดหนึ่ง ซึ่งเขียนเป็นสูตรทั่ว ๆ ไปได้ดังนี้ คือ

X_1 เลือกจากตัวกลางของ p - digit ของ 2 p - digit X_0^2

X_2 เลือกจากตัวกลางของ p - digit ของ 2 p - digit X_1^2

X_3 เลือกจากตัวกลางของ p - digit ของ 2 p - digit X_2^2

X_n เลือกจากตัวกลางของ p - digit ของ 2 p - digit X_{n-1}^2

X_0 = ค่าเริ่มต้นที่ใช้ในการสร้างตัวเลขสุ่ม

วิธีนี้ได้รับการคัดค้านอย่างแข็งขันว่า [3] เลขที่ได้มาตามวิธีการนี้จะเป็นเลขสุ่มได้อย่างไร ในเมื่อจำนวนเลขเหล่านั้นได้มาอย่างมีกฎเกณฑ์จากจำนวนเลขจำนวนก่อนคำตอบในปัญหานี้ก็คือ จริงอยู่จำนวนเลขดังกล่าวนี้ มิใช่จำนวนเลขที่สุ่มขึ้นมาอย่างแท้จริง จำนวนเลขที่ปรากฏขึ้นในชุดนั้น มีคุณลักษณะเหมือนเลขสุ่มทั้งนี้ก็เพราะ ความสัมพันธ์ระหว่างเลขจำนวนแรกกับจำนวนถัดไปนั้น ไม่มีความสำคัญทางกายภาพ (Physical) ให้สังเกตุเห็นได้เลยในทางปฏิบัตินั่นคือ แม้คุณลักษณะของการได้ตัวเลขมานั้นจะไม่ใช่การสุ่ม แต่ตัวเลขเหล่านั้นหาได้มีความสัมพันธ์ต่อกันและกันไม่ ดังนั้นจึงยังพอที่จะเชื่อถือได้ว่า วิธี Middle - square สร้างตัวเลขขึ้นมาอย่างไม่อาจคาดคะเนลำดับก่อนหลังของมันได้เลย

Von Neumann เจ้าของวิธีการ Middle - square ก็ได้ชี้แจงให้เห็นถึงข้อจำกัดของวิธีนี้ไว้เหมือนกัน คือ ชุดลำดับของตัวเลขมีความโน้มเอียงที่จะเกิดขึ้นซ้ำกันเป็นชุด ๆ และช่วงของการเริ่มซ้ำกันนั้นมีขนาดสั้น ๆ ตัวอย่างเช่น ถ้ามีเลข "0" เกิดขึ้นกับจำนวนเลขในชุดลำดับนั้นแล้ว มันจะเกิดอยู่ซ้ำ ๆ เช่นนั้นเรื่อย ๆ ไป

มีบุคคลหลายคนที่ได้ทำการทดลองเกี่ยวกับวิธี Middle - square นี้ เมื่อตอนต้นปี คศ. 1950 โดยทำการทดลองกับจำนวนเลขชุดที่มีตัวเลขจำนวนละ 4 ตัว ดังเช่น การทดลอง

---

3. Donald E. Knuth, The Art of Computer Programming VOL 2. page 2 - 6

ของ G.E. Forsythe ซึ่งทดลองเริ่มต้นด้วย จำนวนเลขที่แตกต่างกัน 16 จำนวน เขาพบว่ามีอยู่ 12 จำนวน ให้ชุดของเลขสุ่มที่มี cycle สิ้นสุดท้ายด้วย 6100, 2100, 4100, 8100, 6100,..... ส่วนอีก 2 จำนวนนั้นโน้มเอียงเข้าหา 0000

N. Metropolis ได้ทำการทดลองวิธี Middle - square นี้ด้วยตัวเลขในระบบฐาน 2 โดยใช้จำนวนเลข 20 bits เขาได้แสดงให้เห็นว่าจะเกิด cycle ต่าง ๆ กันถึง 13 แบบ และ cycle ที่มีช่วงยาวที่สุดนั้นมีความยาวถึง 142 จำนวน และถ้าพบเลขศูนย์เข้ามาปรากฏแล้วก็อาจเปลี่ยนตั้งจำนวนเลขชั้นมาใหม่เพื่อทำการตั้งต้นใหม่ก็ได้

N. Metropolis พบว่าการทดลองทำโดยใช้เลขจำนวน 38 bits นั้นให้ชุดลำดับของจำนวนเลขประมาณ 750,000 จำนวน ก่อนที่จะเกิดความบกพร่องในวิธีการขึ้น และจากผลการทดลองพบว่าจำนวนก่อนที่จะเกิดความบกพร่องในวิธีการขึ้น และจากผลการทดลองพบว่าจำนวน 750,000 x 38 bits ผ่านการทดสอบความสุ่มได้ นั่นคือวิธีการ Middle - square ให้ผลดีพอที่จะนำไปใช้งานได้ แต่จะเชื่อถือเสียจนไม่ยั้งคิดอะไรเลยนั้นไม่ได้ จะเชื่อถือได้ก็ต่อเมื่อได้มีการตรวจสอบการคำนวณดูแล้วอย่างละเอียดถี่ถ้วน

ข้อเสียของ Middle-spuare [4]

1. ชุดเลขลำดับที่ได้ไม่ค่อยยาวนักเมื่อเทียบกับวิธีอื่น ๆ
2. การสร้างตัวเลขสุ่มทำได้ช้ากว่าวิธีอื่น ๆ

2.2 วิธี Power Residue [5]

วิธี Power Residue เป็นวิธีหนึ่งที่ใช้กันอย่างแพร่หลายก่อนที่จะกล่าวถึงวิธี Power Residue เพื่อให้ง่ายต่อการเข้าใจจะขอกล่าวถึงทฤษฎีตัวเลข และแนวความคิดเกี่ยวกับตัวเลขเสียก่อน

---

4. Joe H. Mize and J. Grady Cox., Essential of Simulation page 64
5. Shan S. Kuo; Computer Application of Numerical Methods (Addison - Wesley Publishing Company) page 335 - 342

นิยาม

ถ้ามีตัวเลข 2 จำนวน คือ s และ t ผลต่างของ s และ t หารลงตัวด้วย M แล้ว (M เป็นตัวเลขจำนวนเต็ม) เราเรียก s ว่า "Congruent ต่อ t modulo M" ซึ่งเขียนเป็นสัญญลักษณ์แทนได้ดังนี้ $s = t \bmod M$

s = ค่าบวกที่เล็กที่สุด

t = เลขจำนวนหนึ่ง

M = ตัวเลขจำนวนเต็มตัวหนึ่ง

ถ้าหาก s, t และ M เป็นไปตามนิยามแล้วจะเกิดความสัมพันธ์ขึ้น 2 ประการ คือ

1. เศษของ s/M และ t/M จะเท่ากัน
2. $s - t = i.M$ เมื่อ i คือ integer

t/M = truncated - integer devision

อาศัยแนวความคิดของทฤษฎีเกี่ยวกับตัวเลข เรากำหนดเซทของ Power Residue ของ s และ S_n โดยใช้สัญญลักษณ์แสดงได้ดังนี้

$$s_n = t \pmod M \qquad n = 1, 2, 3 .....$$

เราเรียก s_n ที่ได้นี้ว่า Power Residue โดยอาศัยแนวความคิดของ Power Residue การสร้างตัวเลขสุ่มโดยใช้ binary computer นั้น เราสามารถสร้างตัวเลขสุ่ม U_{n+1} โดยอาศัยความสัมพันธ์ที่รู้จักกันดีคือ

$$U_{n+1} = x U_n \pmod{2^d}$$

เมื่อ

U_{n+1} = ตัวเลขสุ่มตัวต่อไป

U_n = ตัวเลขสุ่มตัวก่อน

d = จำนวน bit ในหนึ่ง word

x = ตัวคงที่ที่ใช้เป็นตัวคูณ

2.3 วิธี Linear Congruence [6]

ในปี 1948 D.H. Lehmer เป็นคนคิดวิธีสร้างตัวเลขสุ่มที่ใช้ได้ผลดีและเป็นที่รู้จักกันมาจนทุกวันนี้ คือวิธี Congruence ซึ่งมีอยู่ 2 แบบ ที่เป็นที่รู้จักกันดี คือ

2.3.1 วิธี Multiplicative

วิธีนี้ใช้ความสัมพันธ์ซึ่งเขียนเป็นรูปสูตรได้ดังนี้

$$X_{i+1} = a\ X_i \pmod m$$

เมื่อ

X_i = ค่าเริ่มต้น $X_i \geq 0$

a = ตัวคูณ $a \geq 0$

m = modulus $m > X_i,\ m > 0$

หมายความว่าเอาค่าเริ่มต้น X_i มาคูณด้วย a ซึ่งเป็นตัวคูณที่คงที่และนำเอาผลที่ได้มา modulus m (หมายความว่า เอาผลที่ได้จากผลคูณของ a และ X มาหารด้วย m เหลือเศษเท่าไร เศษที่ได้คือ $X_i + 1$) ดังนั้นเราจะได้ตัวเลขสุ่มที่อยู่ระหว่าง 0 และ m - 1 ถ้า m คือค่าที่เราเลือกให้เป็นค่าที่มากที่สุด ซึ่งเท่ากับขนาดของ word ของคอมพิวเตอร์ซึ่งเป็นเทคนิคอันหนึ่งที่เราจะได้ เลขลำดับที่มีความยาวสูงสุดและทำได้รวดเร็วยิ่งขึ้น เพราะวิธีหารนั้นคอมพิวเตอร์ทำได้ช้ากว่าการคูณ เราอาจจะใช้วิธีการคูณแทนการหารได้ ซึ่งขึ้นอยู่กับเทคนิคการเขียนโปรแกรม เพื่อช่วยประหยัดเวลาคำนวณของคอมพิวเตอร์

---

6. James R. Emshaff and Roger L. Sission, Design and Use of Computer Simulation models page 175 - 177

2.3.2 วิธี Mixed[7]

วิธีนี้เป็นวิธีสร้างตัวเลขสุ่มที่ใช้ได้ผลดี และเป็นที่รู้จักกันดีวิธีหนึ่ง ซึ่งใช้ความสัมพันธ์ซึ่งเขียนเป็นสูตร ได้ดังนี้

$$X_{i+1} = (a X_i + C) \bmod m \qquad \text{เมื่อ} \quad i \geq 0$$

X_i = ค่าเริ่มต้น $\quad X_i \geq 0$

a = ตัวคูณ $\quad a \geq 0$

c = ตัวเพิ่มที่ละน้อย (increment) $\quad c \geq 0$

m = modulus $\quad m > X_o,\ m > a,\ m > c$

ชุดลำดับตัวเลขที่ได้จากการสร้าง ไม่ได้เป็นเลขสุ่มเสมอไป สำหรับทุกค่าของ X_o, a, c และ m ซึ่งการเลือกค่าต่าง ๆ เหล่านี้จำเป็นต้องมีวิธีเลือกที่ดีจึงจะได้ชุดลำดับของตัวเลขสุ่มที่ยาวตามความต้องการ และเป็นเลขสุ่มจริง ซึ่งจะกล่าวถึงวิธีเลือกค่าต่าง ๆ เหล่านี้ในตอนต่อไป

บางครั้งชุดลำดับตัวเลขที่ได้เป็น Loop การเกิดชุดลำดับตัวเลขที่เป็น loop นี้เป็นคุณสมบัติข้อหนึ่งที่เกิดกับการสร้างตัวเลขสุ่มที่มีสูตรทั่ว ๆ ไปเป็น

$$X_{n+1} = f(X)$$

การเกิดตัวเลขซ้ำ ๆ กันเป็น cycle ของตัวเลขนั้น เราเรียกว่า "คาบ" (period) โดยปรกติชุดลำดับตัวเลขที่จะใช้งานได้จริง ๆ ต้องมีคาบยาวพอสมควร จึงจะใช้ได้

ในปี 1967 Chamber ได้เปรียบเทียบวิธี Congruential ทั้ง 2 แบบไว้พอสรุปได้ดังนี้

---

7. Donald E. Knuth, The Art of Computer Computer Programming (Addison-Wesley Publishing Company) page 9 - 10

1. Mixed method ให้ cycle ที่ยาวกว่า
2. Multiplicative method สามารถสร้างตัวเลขที่ผ่านการทดสอบทางสถิติเกี่ยวกับ random ได้มากกว่า
3. Multiplicative method โดยปรกติจะสร้างตัวเลขสุ่มได้เร็วกว่า

จากการศึกษาถึงวิธีสร้างตัวเลขสุ่มวิธีต่าง ๆ เราพบว่าวิธี Multiplicative เป็นวิธีที่เหมาะที่สุด เพราะผ่านการทดสอบความสุ่มมากที่สุด ดังนั้นการสร้างตัวเลขสุ่ม เราจึงควรเลือกวิธี Multiplicative แต่การเลือกวิธี Multiplicative เป็นวิธีหนึ่งของ Linear congruence แม้ว่าวิธีนี้จะเป็นวิธีที่ดีและเหมาะที่สุดก็ตาม แต่ถ้าเราไม่ทราบเทคนิคการใช้วิธี Multiplicative แล้ว การสร้างตัวเลขสุ่มด้วยวิธีนี้ก็ไม่ให้ผลดีเท่าที่เราต้องการ ดังนั้นเราจึงจำเป็นต้องศึกษาถึงเทคนิคการเลือกค่าต่าง ๆ ที่จะทำให้วิธี Multiplicative ได้ผลดีเท่าที่ควร ซึ่งเราจะศึกษาในบทต่อไป

วิธี Multiplicative เป็นวิธีที่ดีสำหรับการสร้างตัวเลขสุ่ม เรานำเอาสูตร Multiplicative มาเขียนเป็นโปรแกรมคอมพิวเตอร์เพื่อสดวกในการใช้งาน

สูตร Multiplicative Method

$$X_{i+1} = a\ X_i \pmod m$$

เมื่อ

X_i = ค่าเริ่มต้น $X_i \geq 0$

a = ตัวคูณ $a \geq 0$

m = modulus $m > x_i,\ m > a$

จากสูตร วิธี Multiplicative เรานำมาเขียนโปรแกรมป้อนเข้าเครื่องคอมพิวเตอร์ และให้เครื่องคอมพิวเตอร์สร้างชุดลำดับตัวเลขสุ่มออกมา

โปรแกรมคอมพิวเตอร์

ในการเขียนโปรแกรมเราใช้สัญลักษณ์ ดังนี้

ค่าเริ่มต้น (Initial value) = X I

ตัวคูณ (Multiplier) = A

โมดูลัส (Modulus) = Y

```
      C     PROGRAM GENERATING RANDOM NUMBERS.
001           DIMENSION RAND (250)
002           Y=1073741824
003           READ(2,20)A,XI
004        20 FORMAT(2F10.0)
005           DO 22 J=1,4
006           WRITE(3,33)
007        33 FORMAT(1H1/1H3)
010           WRITE(3,35)
011        35 FORMAT(//27X,23HTABLE OF RANDOM NUMBERS,//)
012           DO 55 I=1,250
013           T=A*XI
014           Z=T/Y
015           IZ=Z
016           Q=IZ
017           F=Z-Q
020           XI=F*100000000.
021        55 RAND(I)=F
022           WRITE(3,30)RAND
023        30 FORMAT(5E14.8)
024        22 CONTINUE
025           WRITE(3,40)
026        40 FORMAT(//27X,20HEND OF RNDOM NUMBER,//)
027           STOP
030           END
```