ECE/CS 541
Computer System Analysis:
Intro to Pseuorandom Number Generators

Mohammad A. Noureddine
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Fall 2018

Announcements

- **Keep working on projects!**
  - Worth 35% of the grade
  - Presentation due December 15th, paper due December 17th
  - One more homework

Learning Objectives

- Or what is this course about?

- At the start of the semester, you should have
  - Basic programming skills (C++, Python, etc.)
  - Basic understanding of probability theory (ECE313 or equivalent)

- At the end of the semester, you should be able to
  - Understand different system modeling approaches
    - Combinatorial methods, state-space methods, etc.
  - Understand different model analysis methods
    - Analytic/numeric methods, simulation
  - Understand the basics of discrete event simulation
  - **Design simulation experiments and analyze their results**
  - Gain hands-on experience with different modeling and analysis tools

Motivation

- We need random numbers in our simulations:
  - Arrival times, departure times, completion times, etc.
  - Picking outcomes for stochastic activities
- Focus for today is to examine more closely how such numbers can be obtained and what they really represent

Random Number Generator

- A random number generator (RNG) is a computational or physical device designed to generate a sequence of numbers that appear random.
- By random, it means that they do not exhibit any discernible pattern.
- Stated another way, given a sequence of numbers, the next number in the sequence can not be predicted.

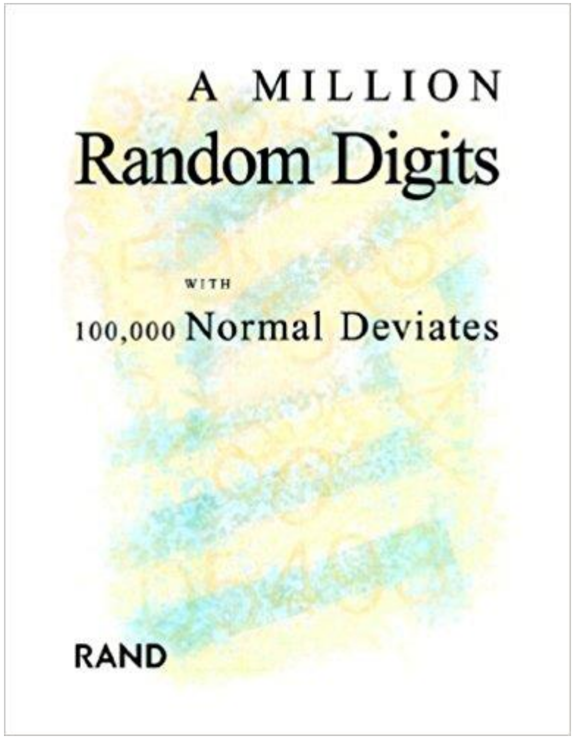
Historical Mechanical Generation Methods

- Historical methods of generating random numbers include:

- These mechanical methods are slow and expensive.

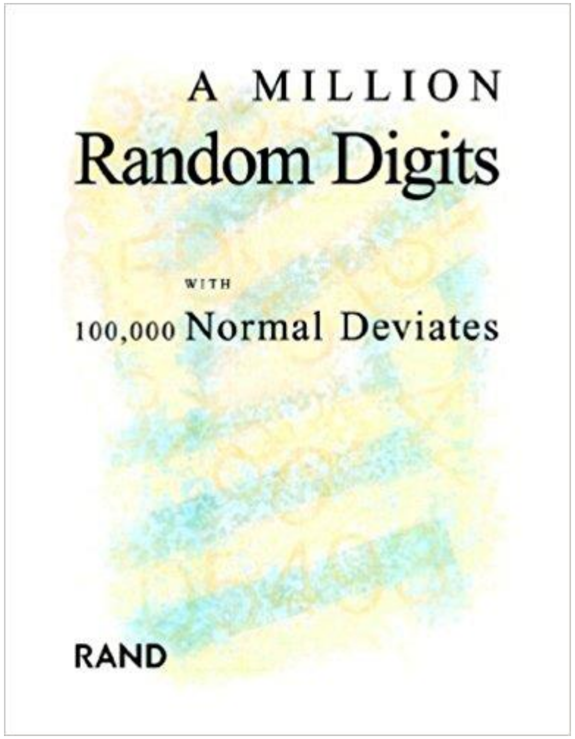
Lookup Tables

- Can instead use a lookup table.
- Famous example: the Rand Coperation’s 1955 book “A Million Random Digits with 100,000 Normal Deviates”



Lookup Tables

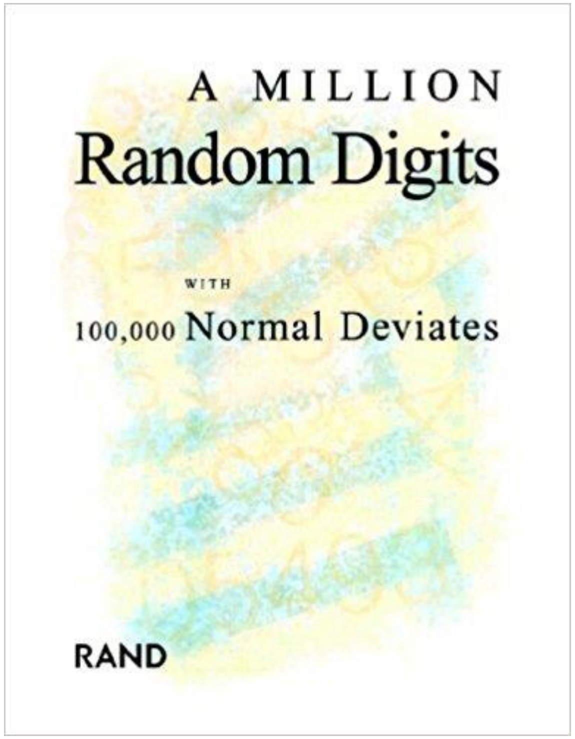
- Can instead use a lookup table.
- Famous example: the Rand Coperation's 1955 book "A Million Random Digits with 100,000 Normal Deviates"



- 4-star review: Such a terrific reference work! But with so many terrific random digits, it's a shame you didn't sort them, to make it easier to find the one you're looking for.

Lookup Tables

- Can instead use a lookup table.
- Famous example: the Rand Coperation's 1955 book "A Million Random Digits with 100,000 Normal Deviates"



- 4-star review: Such a terrific reference work! But with so many terrific random digits, it's a shame you didn't sort them, to make it easier to find the one you're looking for.
- 1-star review: The book is a promising reference concept, but the execution is somewhat sloppy. Whatever generator they used was not fully tested. The bulk of each page seems random enough. However, at the lower left and lower right of alternate pages, the number is found to increment directly.

PRNGs

- It would be useful to have a fast, inexpensive way to produce a stream of numbers for simulation.

- Computers are deterministic, so can't generate random numbers.

- Computers can generate numbers in such a complex manner that, to all intents and purposes, the successive numbers have no discernible pattern, and can be "random enough" for simulation. What we want is a Pseudorandom Number Generator (PRNG).

Desirable Characteristics of a PRNG

- Lehmer: A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence it to be put.

Desirable Characteristics of a PRNG

- Lehmer: A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence it to be put.

- Goal: to produce a sequence of numbers in [0, 1] that simulates, or imitates, the ideal properties of random numbers:

Desirable Characteristics of a PRNG

- Lehmer: A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence it to be put.

- Goal: to produce a sequence of numbers in [0, 1] that simulates, or imitates, the ideal properties of random numbers:

- Important considerations in random number generators:
  - Closely approximate the ideal statistical properties of **uniformity** and **independence**
  - Fast
  - Portable to different computers
  - Replicable

Cryptographically Secure RNG

- Pseudorandom number generators are also used in cryptography

- A *cryptographically secure* pseudorandom number generator (CRNG)
  – Should pass the "next bit" test: Given the first *k* bit of a random sequence, no polynomial time algorithm can predict the next bit with probability of success > 50%
  – Should pass "state compromise extension": in event that some or all of state is compromised, it should be impossible to reconstruct stream of random numbers prior to revelation

Techniques for Generating Random Numbers

- Von Neumann "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

- PRNGs that have been used for simulations include:
  – Least-Squares Method
  – Linear Congruential Method (LCM)
  – Tausworthe Generators
  – Mersenne Twister

Middle-Square Method

- Invented by John von Neumann, presented in 1949.
- Procedure:
  - Take n-digit number, and square it
  - If the result has fewer than 2n digits, add leading zeroes
  - Use the middle n digits as the next number in the sequence
  - Repeat to generated more random numbers

Middle-Square Method

- Invented by John von Neumann, presented in 1949.
- Procedure:
  - Take n-digit number, and square it
  - If the result has fewer than 2n digits, add leading zeroes
  - Use the middle n digits as the next number in the sequence
  - Repeat to generated more random numbers

- Weaknesses
  - Short cycles, e.g. 2916

Demonstration of Least Squares Method

First number is 2916

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: $8503056 \rightarrow 08503056$

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: $8503056 \rightarrow 08503056$
3. Take middle four digits 08**5030**56 $\rightarrow$ 5030

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: $8503056 \rightarrow 08503056$
3. Take middle four digits 08**5030**56 $\rightarrow$ 5030
4. Second number is 5030

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: 8503056 → 08503056
3. Take middle four digits 08**5030**56 → 5030
4. Second number is 5030

1. Square 5030: $5030^2 = 25300900$
2. No need to add leading 0
3. Take middle four digits: 25**3009**00 → 3009
4. Third number is 3009

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: $8503056 \rightarrow 08503056$
3. Take middle four digits 08**5030**56 $\rightarrow$ 5030
4. Second number is 5030

1. Square 5030: $5030^2 = 25300900$
2. No need to add leading 0
3. Take middle four digits: 25**3009**00 $\rightarrow$ 3009
4. Third number is 3009

1. Square 3009: $3009^2 = 9054081$
2. Add leading 0: 09054081
3. Take middle four digits: 09**0540**81 $\rightarrow$ 0540
4. Four number is 0540

Demonstration of Least Squares Method

First number is 2916

1. Square 2916: $2916^2 = 8503056$
2. Add leading 0: 8503056 $\rightarrow$ 08503056
3. Take middle four digits 08**5030**56 $\rightarrow$ 5030
4. Second number is 5030

1. Square 5030: $5030^2 = 25300900$
2. No need to add leading 0
3. Take middle four digits: 25**3009**00 $\rightarrow$ 3009
4. Third number is 3009

1. Square 3009: $3009^2 = 9054081$
2. Add leading 0: 09054081
3. Take middle four digits: 09**0540**81 $\rightarrow$ 0540
4. Four number is 0540

1. Square 0540: $0540^2 = 291600$
2. Add leading 0: 00291600
3. Take middle four digits: 00**2916**00 $\rightarrow$ 2916
4. Fifth number is 2916

Middle-Square Method

- Invented by John von Neumann, presented in 1949.
- Procedure:
  - Take n-digit number, and square it
  - If the result has fewer than 2n digits, add leading zeroes
  - Use the middle n digits as the next number in the sequence
  - Repeat to generated more random numbers

- Weaknesses
  - Short cycles, e.g. 2916

Middle-Square Method

- Invented by John von Neumann, presented in 1949.
- Procedure:
  - Take n-digit number, and square it
  - If the result has fewer than 2n digits, add leading zeroes
  - Use the middle n digits as the next number in the sequence
  - Repeat to generated more random numbers

- Weaknesses
  - Short cycles, e.g. 2916
  - What happens if the first half of the digits are all zeroes?
  - What happens if the middle digits are all zeroes?

Linear Congruential Method

- The Linear Congruential Generator is defined by the recurrence relation

$$X_{n+1} = (aX_n + c) \mod m$$

  - X is the sequence of pseudorandom values
  - m is the modulus
  - a is the multiplier
  - c is the increment
  - X_0 is the seed

- The random integers are being generated in [0, m – 1], so to convert the integers to random numbers

$$R_i = \frac{X_i}{m}, i = 1,2,3 \ldots$$

- The choice of a, c, m, and X_0 affect the statistical properties of the generated numbers

Example LCM

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:
  - $X_1 = (17 * 27 + 43) \bmod 100 = 502 \bmod 100 = 2$, so $R_1 = 0.02$

Example LCM

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:
  - $X_1 = (17 * 27 + 43) \bmod 100 = 502 \bmod 100 = 2$, so $R_1 = 0.02$
  - $X_2 = (17 * 2 + 43) \bmod 100 = 77$ so $R_2 = 0.77$

Example LCM

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:
  - $X_1 = (17 * 27 + 43) \bmod 100 = 502 \bmod 100 = 2$, so $R_1 = 0.02$
  - $X_2 = (17 * 2 + 43) \bmod 100 = 77$ so $R_2 = 0.77$
  - $X_3 = (17 * 77 + 43) \bmod 100 = 1352 \bmod 100 = 52$ so $R_3 = 0.52$

Example LCM

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:
  - $X_1 = (17 * 27 + 43) \bmod 100 = 502 \bmod 100 = 2$, so $R_1 = 0.02$
  - $X_2 = (17 * 2 + 43) \bmod 100 = 77$ so $R_2 = 0.77$
  - $X_3 = (17 * 77 + 43) \bmod 100 = 1352 \bmod 100 = 52$ so $R_3 = 0.52$
  - and so on…

Potential Issues

- Is there a potential problem with cycles? Yes!
- Recall LCG: $X_{n+1} = (aX_n + c) \mod m$
- Example: Let $m = 10$, $a = 4$, $c = 3$, and $x_0 = 9$

Potential Issues

- Is there a potential problem with cycles? Yes!
- Recall LCG: $X_{n+1} = (aX_n + c) \mod m$
- Example: Let m = 10, a = 4, c = 3, and x_0 = 9

- However, the Hull-Dobell Theorem says that, when $c \neq 0$, a LCG has a period equal to m iff:
  1. m and the offset c are relatively prime
  2. a – 1 is divisible by all prime factors of m
  3. a – 1 is divisible by 4 if m is divisible by 4

Potential Issues

- Is there a potential problem with cycles? Yes!
- Recall LCG: $X_{n+1} = (aX_n + c) \mod m$
- Example: Let m = 10, a = 4, c = 3, and x_0 = 9

- However, the Hull-Dobell Theorem says that, when $c \neq 0$, a LCG has a period equal to m iff:
  1. m and the offset c are relatively prime
  2. a – 1 is divisible by all prime factors of m
  3. a – 1 is divisible by 4 if m is divisible by 4

- Modulus calculation can be expensive.
- Speedup: make the modulus a power of 2.

Quality of Numbers

- What makes a good pseudorandom number generator?

Statistical Tests

- Lehmer: A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence it to be put.