Pseudo Random Number Generators

Mohammad Humam Khan - 180123057
Kartikeya Singh - 180123021

SUMMARY

- Problem Addressed
- Random Number Generator
- Pseudo Random Number Generator
- Middle Square Method
- Linear Congruential Generator(LCG)
- Lagged Fibonacci Generator(LFG)
- Mersenne Twister Algorithm
- Middle-Square Weyl Sequence Algorithm
- Conclusion

Problem Addressed

- Can we generate truly random numbers using a Computational Algorithm ?
- If Yes, Then What is the Algorithm?
- If No, upto what extent of randomness we can get using a computational machine ?
- These questions led to evolution of a very important field in study of Algorithms namely Random Number Generators and Pseudo Random Number Generators.

DEFINITIONS

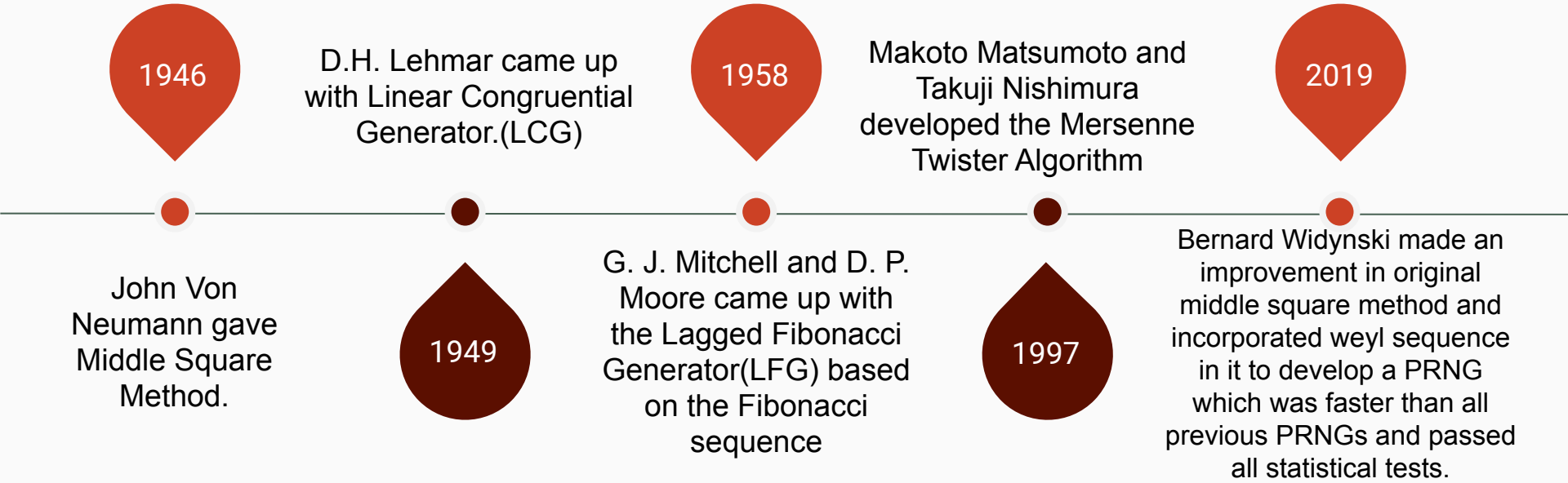
RANDOM NUMBER GENERATOR

A random number generator (RNG) is a computational mathematical algorithm that is designed to generate a random sequence of numbers that should not display any distinguishable patterns in their appearance or generation.

PSEUDO RANDOM NUMBER GENERATOR

- A pseudorandom number generator, also known as a deterministic random bit generator, is an algorithm for generating a sequence of numbers whose properties resemble the properties of sequences of random numbers.
- The PRNG-generated sequence cannot be said to be truly random, because it is determined by an initial value, called the PRNG's seed.
- The seed may be any number, but it usually comes from seconds on a computer system's clock.

Chronology


1946
John Von Neumann gave Middle Square Method.
1949
D.H. Lehmar came up with Linear Congruential Generator.(LCG)
1958
G. J. Mitchell and D. P. Moore came up with the Lagged Fibonacci Generator(LFG) based on the Fibonacci sequence
1997
Makoto Matsumoto and Takuji Nishimura developed the Mersenne Twister Algorithm
2019
Bernard Widynski made an improvement in original middle square method and incorporated weyl sequence in it to develop a PRNG which was faster than all previous PRNGs and passed all statistical tests.

Middle Square Method

ALGORITHM

1. Take a random number.
2. Square it.
3. Take middle digits of resulting number as the result.
4. Use this number as seed for next iteration.

EXAMPLE

1. Consider $x_0 = 643794$
2. Then $x^2 = 414\mathbf{470714}436$
3. Then result i.e. $x_1 = 470714$.
4. For next iteration x = *470714*
5. Then $x^2 = 221\mathbf{571669}796$
6. Then result i.e. $x_2 = 571669$
7. Hence sequence of random numbers is: *(643794, 470714, 571669, …)*

Linear Congruence Generator(LCG)

ALGORITHM

1. Take an integer (X_0) as the seed.
2. Take some integers a, c, m which are constants.
3. The desired sequence is obtained by the relation $X_{n+1} = (aX_n + c) \bmod m$.

EXAMPLE

1. Take $X_0 = 1$, $a = 3$, $c = 2$ and $m = 7$.
2. $X_1 = (3*1 + 2) \bmod 7 = 5$.
3. $X_2 = (3*5 + 2) \bmod 7 = 3$.
4. $X_3 = (3*3 + 2) \bmod 7 = 4$.
5. The sequence generated is given as *(1,5,3,4,0,2 ...)*.

Lagged Fibonacci Generator(LFG)

ALGORITHM

1. Take integers *j, k (k > j > 0)* and *m*.
2. Initial values of $X_0, X_1, \ldots X_{k-1}$ are supplied.
3. The desired sequence is obtained by the relation $X_n = (X_{n-k} * X_{n-j}) \bmod m$. Where * is any binary operator

EXAMPLE

1. Take $j = 1, k = 2, m = 8$.
2. Take $X_0 = 3$, $X_1 = 4$ and the binary operation to be addition.
3. $X_2 = (3 + 4) \bmod 8 = 7$
4. $X_3 = (4 + 7) \bmod 8 = 3$
5. The resultant sequence is *(3,4,7,3,2,5,7...)*

Mersenne Twister Algorithm

- The Mersenne Twister is a pseudorandom number generator (PRNG) with its period length chosen to be a Mersenne prime.
- Mersenne prime is a prime number that is one less than a power of two i.e it is a prime number of the form $M_n = 2^n - 1$ for some integer n.
- The Mersenne Twister algorithm is based on a matrix linear recurrence over a finite binary field F_2
- For a w-bit word length, the Mersenne Twister generates integers in the range $[0, 2^w-1]$.
- For a particular choice of parameters, the algorithm generates very large period of $2^{19937} - 1$ and 623-dimensional equidistribution up to 32-bit accuracy, while using a working area of only 624 words.

Middle Square Weyl Sequence

ALGORITHM

1. Take a random number $X = X_0$ as seed.
2. Square X_0 i.e. $X := X_0^2$.
3. After squaring 64-bit integer we will get 128-bit int but since x is a 64-bit integer, only least significant 64-bits of resulting number will be stored in x.
4. Calculate next weyl sequence number w+=s.
5. Then add a weyl sequence number w to it i.e. *X += w.* The most significant 32-bits of resulting X is the desired result.
6. The operation (x>>32) | (x<<32) flips first 32 bits with last 32 bits and the last 32 bits are stored in x and returned by the function. (<< and >> represent the bit shifting operators.)

C Code

```c
#include <stdint.h>
uint64_t x = 0, w = 0,
s = 0xb5ad4eceda1ce2a9;

inline static uint32_t msws() {
    x *= x; // squaring
    w += s; // weyl sequence
    x += w; // adding weyl sequence
    return x = (x>>32) | (x<<32);
}
```

Middle Square weyl Sequence Example

- Take *x = 0xace983fe671dbd09* i.e. a 64-bit number , *s = 0xb5ad4eceda1ce2a9* and *w = 0*.
- Then binary representation of x is:
  1010110011101001100000111111111001100111000111011011110100001001
- When this number is squared, it becomes the 128-bit number *0x74ca9e5f63b6047f6a65456d9da04a51*.
- But since x is a 64-bit number only 64 LSBs are stored in x, then x is actually equal to *0x6a65456d9da04a51*.
- On adding the weyl sequence (*w = 0xb5ad4eceda1ce2a9*) to x we get x = *0x2012943c77bd2cfa*
- *Then performing last step of algorithm for extracting middle 32 bits of x we get: (x>>32)|(x<<32) = 0x77bd2cfa2012943c*
- *The function will return Least Significant 32-bits as result i.e. 0x2012943c*
- *This is the 32-bit middle that is our next random number in sequence.*

CONCLUSION

- No True Random Number Generating Algorithm has been found till date.
- However there has been significant progress in the field of Pseudorandom Number Generation.
- We saw a few important Pseudo-Random Number Generating Algorithms such as Middle Square Weyl Sequence Generator, Mersenne Twister, Lagged Fibonacci Generator, Linear Congruence Generator and Middle Square Generator.

“Anyone who considers arithmetical methods of producing random numbers is of course in a state of sin”

- John Von Neumann

References

We have used following sources for making our presentation:

- https://en.wikipedia.org/wiki/Pseudorandom_number_generator (Definition only)
- https://arxiv.org/pdf/1704.00358v4.pdf (Middle Square Weyl Sequence)
- https://mcnp.lanl.gov/pdf_files/nbs_vonneumann.pdf (von Neumann Middle Square Method)
- https://dl.acm.org/doi/pdf/10.1145/272991.272995 (Mersenne Twister)

Thank You !