Donald Knuth – The Art of Computer Programming

- http://www-cs-faculty.stanford.edu/~knuth/
- **"Look at the subroutine library of each computer installation in your organization, and replace the random number generators by good ones. Try to avoid being too shocked at what you find."**
- **Many random number generators have been written, most of them have demonstrably non-random characteristics, some are embarrassingly bad.**

Minimal Standard

- Multiplicative linear congruential generator with multiplier 16807 and prime modulus $2^{31} – 1$.
- Objective: a virtually infinite sequence of statistically independent random numbers uniformly distributed between 0 and 1.
- D.H. Lehmer produced an algorithm in 1951 that is satisfactory under some conditions.

Lehmer's Algorithm

(i) *modulus*: m—a large *prime* integer

(ii) *multiplier*: a—an integer in the range $2, 3, \ldots, m-1$

and the subsequent generation of the integer sequence $z_1, z_2, z_3 \ldots$ via the iterative equation

(iii) $z_{n+1} = f(z_n)$ for $n = 1, 2, \ldots$

where the *generating function* $f(\cdot)$ is defined for all z in $1, 2, \ldots, m-1$ as

(iv) $f(z) = az \bmod m$.

The sequence of z's must be initialized by choosing an *initial seed* z_1 from $1, 2, \ldots, m-1$. And, as an additional step, the sequence is conventionally normalized to the unit interval via division by the modulus to produce the real sequence $u_1, u_2, u_3, \ldots$ where

(v) $u_n = z_n/m$ for $n = 1, 2, \ldots$

- Two parameters, a and m.
- "Prime Modulus Mulitplicative Linear Congruential Generator"
- Initial seed
- deterministic versus random
- full period versus small period behavior
- The Mersenne Prime

  $2^{31} - 1 = 2147483647$

Three Issues for Implementing a Lehmer Generator

1. Full period periodicity
2. Randomness
   - See Chapter 4
3. Implementation
   - evaluating f(z) = az mod m efficiently and correctly for all values

The Multiplier

The multiplier $a = 7^5 = 16807$ was first suggested by Lewis, Goodman and Miller in 1969 [27], based largely on the fact that

$f(z) = 168072 \bmod 2147483645$

is a full period generating function.

Implementation

- Maxint and oveflows
- Check for correctness and not randomness

Conclusions

- Many computer scientists will never have more than an occasional need to use a random number generator.
- And on those occasions the statistical goodness of the random numbers they generate may not be of paramount importance.
- For some of us, however, convenient and frequent access to a verifiably good random number generator is fundamentally important.
- If you have need for a random number generator, particularly one that will port to a wide variety of systems, and if you are not a specialist in random number generation and do not want to become one, use the minimal standard.
- *It should not be presumed that it is easy to write a better one.*