LA-UR-11-04858

Approved for public release;
distribution is unlimited.

Title: Testing MCNP Random Number Generators

Author(s): Yasunobu Nagaya & Forrest B. Brown

Intended for: MCNP References,
historical document from 2002 unpublished memos

Los Alamos
NATIONAL LABORATORY

research note
Applied Physics Division
Diagnostics Applications
Group X-5, MS F663
Los Alamos, New Mexico 87545

To/MS: Forrest B. Brown
From/MS: Yasunobu Nagaya / X-5, MS F663
Phone/Email: 5-3914 / nagaya@lanl.gov
Symbol: X-5:YN-02-XXX
Date: August 2, 2002

Testing MCNP random number generators

Y. Nagaya and F. B. Brown

ABSTRACT

Linear congruential random number generators (LCGs) are most widely used for particle-transport Monte Carlo methods and most Monte Carlo codes employ 47- or 48-bit LCGs. Recent progress of computers makes the period of the generators shorter. Thus, we picked up possible candidates of 63-bit LCGs and tested the LCGs including the current MCNP random number generator. We performed the spectral test, Knuth's standard tests and Marsaglia's DIEHARD tests for the MCNP generator, 63-bit LCGs extended from the MCNP generator and 63-bit LCGs proposed by L'Ecuyer. We found that the MCNP generator fails some tests in the DIEHARD test suite and the 63-bit LCGs extended from the MCNP RNG fail the spectral test. On the other hand, L'Ecuyer's 63-bit LCGs pass all the tests and their multipliers are excellent. It is considered that they are the most promising LCGs that can be easily upgraded from the current LCG.

1 Introduction

It is needless to say that random number generators (RNGs) play a very important role in Monte Carlo simulation. If the quality of a RNG used in the simulation is poor, we cannot trust all results obtained from the simulation. Thus, RNGs used in the simulation must have robust theoretical properties and must be thoroughly verified with tests.

In general, random numbers generated on computers are called “pseudo” random numbers and the sequence of the numbers has a period or cycle length because the available bit length is limited. RNGs should have a period long enough for the simulation. The period can be known theoretically and an appropriate parameter set must be chosen to achieve the long period.

Another requirement for RNGs is that random numbers must be randomly and uniformly distributed in a certain interval. This is often examined by RNG tests with random numbers actually generated. There are a large number of tests proposed for this purpose and some tests have been used as de facto standard. RNGs used should pass some tests for verification of randomness and uniformity.

Linear congruential generators (LCGs) are most frequently used in Monte Carlo simulation. The LCG is one of the classical generators proposed by Lehmer[1]. A lot of other generators have been proposed and some of them have a longer period than the LCG. Nevertheless, most Monte Carlo codes for particle transport have conventionally used them for a long time. It is because LCGs have the following desirable properties;

1. The sequence is deterministic so that repeated calculations will produce identical results.

2. They are very fast, involving only a small number of arithmetic operations.

3. Initialization is trivial, and the state information to specify the sequence for a history is small (1 word).

4. A simple algorithm exists for skipping ahead to any given point in the random sequence.

5. If 48 bits of precision are used in the LCG, the period is large ($2^{46} \sim 7.0 \times 10^{13}$, or $\sim 10^{14}$) and serial correlation is entirely negligible.

6. The algorithm is robust, that is, it cannot fail.

Most Monte Carlo codes use 47- or 48-bit LCGs that have the modulus of 2^{47} or 2^{48} and the period of 2^{45} or 2^{46}, respectively. The modulus is usually restricted by integer precision of compilers and chosen as a nearly maximum value of available integers for a long period. Such LCGs generate a random number sequence of a period long enough for ordinary Monte Carlo calculations. For example, the current version of MCNP (Version 4) uses a 48-bit LCG and 152917 random numbers are kept for each particle (stride). Then the number of tracked particles from just a sequence is approximately $2^{46}/152917 = 4.6 \times 10^8$.

Recently it is, however, not unusual to perform a calculation for 10^8 histories or more as the computer speed increases rapidly. Even if all random numbers in a sequence are exhausted, the calculation result would be still reliable in most cases but it may cause unpredictable correlation. Therefore, LCGs with a longer period have been recently required. Fortunately, recent most compilers allow to use 64-bit integers and thus we can extend the period easily.

A new RNG package upgraded for MCNP Version 5 (MCNP5) includes not only the original MCNP 48-bit LCG but also several 63-bit LCGs. The 63-bit LCGs have the period of $2^{61}(= 2.3 \times 10^{18})$ and $2^{63}(= 9.2 \times 10^{18})$ for multiplicative and mixed LCGs, respectively. Some 63-bit LCGs in the package are recommended by L'Ecuyer[2] and the others are obtained by slightly changing the parameters to determine LCGs. Therefore, they are subject to the RNG tests.

In this work, all the proposed RNGs for MCNP5 are tested with the standard test suite summarized by Knuth[3] and the DIEHARD test suite proposed by Marsaglia[4].

2 Linear Congruential Generator

2.1 Review of principle and features

The basic recursive equation for the linear congruential generators (LCGs) is given by

$$S_{n+1} = (gS_n + c) \bmod m, \tag{1}$$

where S_n is the integer in the interval $[0, m-1]$, m the modulus $(m > 0)$, g the multiplier $(0 \leq a < m)$, c the increment $(0 \leq c < m)$. Then, the random number ξ_n between 0 and 1 is generated by the following equation;

$$\xi_n = S_n/m. \tag{2}$$

We denote the above LCG as LCG(g, c, m). The LCGs are categorized into 2 types; multiplicative LCGs for $c = 0$ and mixed LCGs for $c \neq 0$.

Apparently the integers generated by Eq. (1) lie between 0 and $m - 1$. Thus the possible maximum period is m. In the case of multiplicative LCGs, the integers lie between 1 and $m - 1$ because $S_i = 0$ cannot be allowed. The possible maximum period is $m - 1$.

The maximum period cannot be achieved for all the sets of (g, c, m, S_0). Our most concern is to find the sets that enable LCGs have the maximum period. For this purpose, we use the following theorems for mixed and multiplicative LCGs, respectively.

Theorem A (See [3, p. 17]) The LCG(g, c, m) has the maximum period m if and only if

1. c is relatively prime to m;
2. $g - 1$ is a multiple of p, for every prime p dividing m;
3. $g - 1$ is a multiple of 4, if m is a multiple of 4.

Theorem B (See [16, p. 592]) The LCG$(g, 0, m)$ has the maximum period $m - 1$ if and only if

1. m is a prime number;
2. g is a primitive root of m.

g is a primitive root of m (prime) if and only if

- $g^{m-1} \equiv 0 \pmod{m}$;
- For all integers $i < m - 1$, the quantity $(g^i - 1)/m$ is not an integer.

Theorems A and B give us to choose the sets of the parameters but there are still a huge number of choices that satisfy Theorem A. What we have to consider first is often the choice of a modulus m. It is restricted by integer precision available on a computing platform. Currently, a type declaration

INTEGER(8) is available on most platforms and the modulus is often less than or equal to 2^{64} in this case.

There are two major choices for the modulus. One is a prime modulus. In particular, a Mersenne prime that has the form of $2^\alpha - 1$ is often used. Such RNGs are often seen in scientific subroutine libraries. The other choice is the modulus of the power of 2. This is also often used because of the computational advantage. However RNGs with such moduli have the following drawbacks;

- They does not have the maximum period $m - 1$ because they does not satisfy Theorem B-1.
- The $(r+1)$-th most significant bit has period length at most 2^{-r} times that of the most significant bit [2].

In spite of these drawbacks, The RNGs with moduli of the power of 2 is traditionally used in Monte Carlo codes for particle transport. We also investigate only those RNGs in this work. For the RNGs, Theorems A for mixed RNGs can be rewritten as follows.

Theorem C (See [16, p. 601]) The LCG$(g, c, 2^\beta)$ has the maximum period 2^β if and only if

1. $g \equiv 1 \pmod 4$;
2. c is odd.

On the other hand, we use the following theorem for multiplicative LCGs instead of Theorem B.

Theorem D (See [16, p. 598]) The LCG$(g, 0, 2^\beta)$ has the maximum period $2^{\beta-2}$ if and only if

1. $g \equiv \pm 3 \pmod 8$;
2. S_0 is an odd integer.

Furthermore, multipliers of the form $A \equiv 5 (\mathrm{mod}\ 8)$ produce more uniformly distributed random numbers than multipliers of the form $A \equiv 3 (\mathrm{mod}\ 8)$ (See [16, p. 600]). We may choose the of the form $A \equiv 5 (\mathrm{mod}\ 8)$ though it is not particularly serious for large β.

We have to find the sets of the parameters that satisfy Theorem C or D at least.

2.2 New MCNP RNGs

A new random number package for MCNP5 includes the following RNGs.

1. $\mathrm{LCG}(5^{19}, 0, 2^{48})$: current MCNP RNG
2. $\mathrm{LCG}(5^{19}, 0, 2^{63})$: multiplicative LCG
3. $\mathrm{LCG}(5^{23}, 0, 2^{63})$: multiplicative LCG
4. $\mathrm{LCG}(5^{25}, 0, 2^{63})$: multiplicative LCG
5. $\mathrm{LCG}(5^{19}, 1, 2^{63})$: mixed LCG
6. $\mathrm{LCG}(5^{23}, 1, 2^{63})$: mixed LCG
7. $\mathrm{LCG}(5^{25}, 1, 2^{63})$: mixed LCG
8. $\mathrm{LCG}(3512401965023503517, 0, 2^{63})$: L'Ecuyer's table
9. $\mathrm{LCG}(2444805353187672469, 0, 2^{63})$: L'Ecuyer's table
10. $\mathrm{LCG}(1987591058829310733, 0, 2^{63})$: L'Ecuyer's table
11. $\mathrm{LCG}(9219741426499971445, 1, 2^{63})$: L'Ecuyer's table, mixed LCG
12. $\mathrm{LCG}(2806196910506780709, 1, 2^{63})$: L'Ecuyer's table, mixed LCG
13. $\mathrm{LCG}(3249286849523012805, 1, 2^{63})$: L'Ecuyer's table, mixed LCG

The first RNG is a 48-bit LCG that has been used for MCNP. This LCG is proposed by Beyer (See [12]) and its validity has been well established through many production runs. The other RNGs that are newly implemented for MCNP5 are 63-bit LCGs. Of course, 64-bit LCGs can be easily realized on current 64-bit based platforms but there are still machine/compiler quirks with a sign bit. Therefore, the 63-bit LCGs are chosen for portability.

LCGs $2 \sim 4$ are 63-bit multiplicative LCGs. LCG 2 has the same multiplier as the original MCNP RNG and is a very good candidate for a 63-bit LCG. However, the multiplier may be slightly small for a modulus 2^63. The most significant bit of 5^{19} is 45 since

$$\begin{aligned} 5^{19} &= 10001010110001110010001100000100100010011110 1_2 \\ &= 2^{44} + 2^{40} + 2^{38} + 2^{36} + 2^{35} + 2^{31} + 2^{30} + 2^{29} + 2^{26} + 2^{22} + 2^{21} \\ &\quad + 2^{15} + 2^{12} + 2^{8} + 2^{5} + 2^{4} + 2^{3} + 2^{2} + 2^{0}. \end{aligned}$$

Thus the first 19 bits are 0's in the 64-bit representation. It does not always lead to the non-randomness of a sequence but it is desirable that each of 64 bits should be randomly arranged with 0 and 1.

The multipliers 5^{23}, 5^{25} and 5^{27} are possible candidates. One reason is that multipliers of odd powers of 5 always 5 modulo 8. Since

$$5^{2i-1} = 5 \times (3 \times 8 + 1)^{i-1} \equiv 5 \pmod 8$$

for $i > 1$, the multipliers of 5^{2i-1} satisfy Theorem D-1. The other reason is that the multipliers can be expressed in the precision of a FORTRAN type declaration INTEGER(8) whose range is $[-2^{63}, 2^{63} - 1]$. However, 5^{27} is rejected from the candidates because of its bit pattern. The following is the bit patterns for 5^{23}, 5^{25} and 5^{27};

$$\begin{aligned}
5^{23} &= 101010010110100000010110001111110000101001010111101101_2 \\
5^{25} &= 100001000101100101010001011000010100000000010100100001 \\
&\quad\ 00101_2 \\
5^{27} &= 11001110110010111000111100100111111101\underline{00001000000000}11 \\
&\quad\ 110011101_2 .
\end{aligned}$$

One can see a regular bit pattern in the underlined part.

LCGs 5 $\sim$ 7 are 63-bit mixed LCGs. The multipliers are the same as those of the multiplicative LCGs. They also satisfy Theorem C-1 since

$$5^{2i-1} = (4+1)^{2i-1} \equiv 1 \pmod 4.$$

The period of the mixed LCGs is 2^{63} and is slightly longer than that of the multiplicative LCGs.

LCGs 8 $\sim$ 13 are 63-bit LCGs proposed by L'Ecuyer [2]. They have a good lattice structure and are recommended to use as RNGs for computer simulation.

3 Tests for RNGs

There are a lot of tests to assess the RNGs. Here, we summarize the tests focusing on those we have used in this work.

The tests are classified into following two categories.

- Theoretical tests: Analyzing the algorithm of RNGs based on the number theory and the theory of statistics.

- Empirical tests: Analyzing the uniformity, patterns and so on of RNs generated by RNGs.

The theoretical tests provide us a clue for a good choice of the RNG parameters such as multiplier, increment, modulus etc. On the other hand, the empirical tests uses output RNs that are used actually, and thus they are useful to verify the algorithm implemented in the program.

The empirical tests can be further classified into some categories.

- Standard tests

- Bit level tests

- Physical tests

In this work, we have performed the standard and Bit level tests with the SPRNG[17] and DIEHARD[4] test routines. The tests used in this work are briefly described in the following sections.

Some of these tests are applied directly to a real-valued sequence of RNs

$$\xi_0, \xi_1, \xi_2, \cdots. \tag{3}$$

However, other tests must be applied to a sequence of random integers. In this case, the sequence of random integers

$$I_0, I_1, I_2, \cdots \tag{4}$$

is obtained from the following rule;

$$I_n = \lfloor d\xi_n \rfloor, \tag{5}$$

where d is an arbitrary integer and $\lfloor x \rfloor$ is the floor of x, that is, the greatest integer such that $\max_{k \leq x} k$. d is sometimes chosen as a power of 2;

$$d = 2^m, \tag{6}$$

where m is an integer. For $0 \leq \xi_n < 1$, ξ_n can be expressed as the following form;

$$\xi_n = b_1 * 2^{-1} + b_2 * 2^{-2} + \cdots + b_{m-1} * 2^{-m+1} + b_m * 2^{-m} + \cdots. \tag{7}$$

Then I_n turns out to be

$$I_n = b_1 * 2^{m-1} + b_2 * 2^{m-2} + \cdots + b_{m-1} * 2^1 + b_m * 2^0. \tag{8}$$

Therefore, I_n represents the m most significant bits of the binary representation of ξ_n.

3.1 Theoretical Test

One of the most useful theoretical tests for LCGs is the spectral test. This test inspects the property of the full period of a RNG. All RNGs currently known to be bad fail the test [3, p. 93].

This test was originally introduced by Coveyou and MacPherson [5] and improved by Dieter [6] and Knuth [7]. Hopkins proposed a revised algorithm with a source program to perform the spectral test [8].

3.1.1 Spectral Test

It is well known that LCGs have regular patterns (lattice structures) when overlapping t-tuples of a random number sequence are plotted in a hypercube [9]. In other words, all the t-tuples are covered with families of parallel $(t-1)$-dimensional hyperplanes. The spectral test determines the maximal distance between adjacent parallel hyperplanes. As one can easily find, the smaller the distance is, the better the RNG is.

Now we define the i-th overlapping t-tuples;

$$(\xi_i, \xi_{i+1}, \cdots, \xi_{i+t-1}) \quad \text{for } t \geq 1,$$

where ξ_i is the i-th random number of a sequence. We regard the t-tuples as a point in the t-dimensional unit hypercube $[0, 1)^t$ If the period of the sequence is M, we can plot M points in the hypercube. Then, there exist multiple families of of parallel $(t-1)$-dimensional hyperplanes that covers all the points. Let $d_t(m, g)$ be the maximal distance between the adjacent parallel hyperplanes. (Recall that m is the modulus and g the multiplier.) The distance is also rewritten as follows [3, p. 94];

$$d_t(m, g) = \frac{1}{\nu_t(m, g)}, \tag{9}$$

where $\nu_t(m,g)$ is called the t-dimensional accuracy of the RNG and defined as follows [3, p. 101];

$$\nu_t(m,g) = \min_i \left\{ \sqrt{\sum_{k=1}^{t} S_{i+k-1}} \;\middle|\; \sum_{k=1}^{t} g^{i-1} S_{i+k-1} \equiv 0 \mod m \right\} \tag{10}$$

for $2 \le t \le T$, given T. The spectral test calculates $\nu_t(m,g)$ and an algorithm is described in Reference [3, p. 101].

There is a theoretical upper bound on $\nu_t(m,g)$ given by

$$\nu_t(m,g) \le \gamma_t^{1/2} \tau^{1/t} \stackrel{\text{def}}{=} \nu_t^*(m), \tag{11}$$

where τ is the number of points per unit volume and γ_t is Hermite's constant. The constant is known for $t \le 8$ (See [10, p. 332]):

$$\begin{aligned} \gamma_1 = 1,\ \gamma_2 = \left(\frac{4}{3}\right)^{1/2},\ \gamma_3 = 2^{1/3},\ \gamma_4 = 2^{1/2}, \\ \gamma_5 = 2^{3/5},\ \gamma_6 = \left(\frac{64}{3}\right)^{1/6},\ \gamma_7 = 4^{3/7},\ \gamma_8 = 2. \end{aligned} \tag{12}$$

Since we consider multiplicative LCGs with modulus 2^β and mixed LCGs with a full period, τ is equivalent to M ($\tau = M$):

$$M = \begin{cases} \frac{m}{4} & \text{for multiplicative LCGs (modulus } 2^\beta) \\ m & \text{for mixed LCGs.} \end{cases} \tag{13}$$

Then the inequality (11) can be rewritten as

$$\nu_t(m,g) \le \gamma_t^{1/2} M^{1/t} \stackrel{\text{def}}{=} \nu_t^*(m). \tag{14}$$

Identically, there is a lower bound on $d_t(m,g)$:

$$d_t(m,g) \ge \gamma_t^{-1/2} \tau^{-1/t} \stackrel{\text{def}}{=} d_t^*(m). \tag{15}$$

In our case, the above inequality can be rewritten as

$$d_t(m,g) \ge \gamma_t^{-1/2} M^{-1/t} \stackrel{\text{def}}{=} d_t^*(m). \tag{16}$$

The normalized maximal distance is often used as a measure and is defined as

$$S_t(m,g) = \frac{d_t^*(m)}{d_t(m,g)}. \tag{17}$$

$S_t(m, g)$ lies between 0 and 1.

Note that the increment c does not appear in the above discussion. In theory, c does not affect the spectral test [3, p. 97], for $c \neq 0$. However, c affects the results of the spectral test implicitly in our work because we consider the LCGs with modulus 2^β and the existence of c increases the period of them.

There are some criteria to rank LCGs. Knuth proposed a measure $mu_t(m, g)$ that indicates the effectiveness of the multiplier g [3, p. 105]:

$$\mu_t(m, g) = \frac{\pi^{t/2} \nu_t^t(m, g)}{(t/2)! M}, \tag{18}$$

where

$$\left(\frac{t}{2}\right) = \left(\frac{t}{2}\right)\left(\frac{t}{2} - 1\right) \cdots \left(\frac{1}{2}\right) \sqrt{\pi} \text{ for } t \text{ odd.} \tag{19}$$

Knuth also introduced a criterion with $\mu_t(m, g)$ as summarized in Table 1.

Table 1: Knuth's criterion for the spectral test

$\mu_t(m, g)$ for $2 \leq t \leq 6$	Result
$\mu_t(m, g) \geq 1$	Pass and the multiplier is excellent.
$1 \geq \mu_t(m, g) \geq 0.1$	Pass.
$0.1 > \mu_t(m, g)$	Fail.

Fishman employed $S_t(m, g)$ to screen multipliers in his papers [11], [12]. He proposed the following criterion;

$$M_T(m, g) \stackrel{\text{def}}{=} \min_{2 \leq t \leq T} S_t(m, g) \geq S, \tag{20}$$

where S is between 0 and 1 and he chose $S = 0.8$. According to his study [12], any multiplier that satisfies the above condition does not exceed $d_t^*(m)$ by more than 25%.

L'Ecuyer also employed same criterion as above to obtain the best multipliers for 31-bit and 15-bit LCGs [13]. Recently, he performed an extensive study to find LCGs of different sizes with good lattice structures and investigated $d_t(m, g)$ for higher dimensions [2]. In the paper, he employed extended criteria $M_8(m, g)$, $M_{16}(m, g)$ and $M_{32(m,g)}$ and proposed the best multiplier for each criterion.

3.2 Standard Tests

The standard tests have been used widely to check the quality of RNGs and were well reviewed by Knuth[3].

3.2.1 Equidistribution test (Frequency test)

The equidistribution test is a very fundamental test for Monte Carlo calculations. This test check whether RNs are generated uniformly between 0 and 1. In this test, the RNs can be submitted directly to the Kolmogorov-Smirnov (K-S) test[3] but the chi-square (χ^2) test can be also applied for the random integers. In the latter case, RNs in the interval $[0, 1)$ are multiplied by d and truncated to integers in the interval $[0, d)$. If the RNs are uniformly generated, each integer must have the equal probability $1/d$.

The equidistribution test in the SPRNG routines uses the latter scheme. In addition, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.2 Serial test

This test checks serial correlation of a RN stream. Generally, n groups of k-tuples are comprised of $k * n$ random integers in $[0, d-1]$, and then it is checked whether the k-tuples are uniformly distributed in the k-dimensional hypercube. Each k-tuple must occur with the probability $1/d^k$ unless the serial correlation exists.

The serial test in the SPRNG routines can be used only for pairs of RNs, that is, $k = 2$. We generate n pairs of integers such as (I_1, I_2), (I_3, I_4), $\cdots$, (I_{2n}, I_{2n+1}) and count the number of times that each pair occurs. Each of the d^2 pairs should be equally likely to occur. Thus we apply the chi-square test to these d^2 bins with probability $1/d^2$ in each bin. In addition, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.3 Gap test

In this test, the lengths of "gaps" between random numbers in a certain range are counted. The range is defined with 2 real numbers a, b such that $0 \leq a < b \leq 1$. Suppose that random numbers ξ_j and ξ_r lie between a and b

and others $\xi_{j+1}, \cdots, \xi_{r-1}$ do not; $\underline{\xi_j}, \xi_{j+1}, \cdots, \xi_{r-1}, \underline{\xi_r}$. Then the gap length is r.

As an example, suppose that we get the following RN sequence and set $(a, b) = (0.4, 0.6)$;

$$\begin{array}{l} 0.10574,\ 0.66509,\ \underline{0.46622},\ 0.93925,\ 0.26551,\ 0.11361,\ 0.25714,\ \underline{0.45412}, \\ 0.13971,\ \underline{0.59733},\ 0.26273,\ 0.09937,\ 0.94662,\ 0.14760,\ 0.34662,\ 0.93293, \\ 0.08641,\ 0.02030,\ \underline{0.45855},\ 0.82829,\ 0.20008,\ 0.32121,\ 0.72824,\ \underline{0.45938}, \\ \cdots, \end{array}$$

then we obtain the gap lengths 5, 2, 10, 5, $\cdots$, in turn.

In SPRNG, n gap lengths are counted and gap lengths greater than t is lumped together in a category. The chi-square test is applied to the $t + 1$ categories. In addition, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.4 Poker test (Partition test)

We generate n groups of k successive random integers (k-tuples) in $[0, d-1]$ and count the number of distinct integers in each k-tuple. A chi-square test is then applied to the k categories.

Suppose that we consider the following random integer sequence for $d = 5$,

$$0, 3, 2, 4, 1, 0, 1, 2, 0, 2, 1, 0, 4, 1, 1, 4, 0, 0, 2, 4, \cdots$$

and make 5-tuples ($k = 5$). Then, we obtain the following result.

5-tuple	distinct integers	hand
(0, 3, 2, 4, 1)	5	all different
(0, 1, 2, 0, 2)	3	two pair
(1, 0, 4, 1, 1)	3	three of a kind
(4, 0, 0, 2, 4)	3	two pair
$\cdots$		

The above example shows the simple case of the classical poker test. In this example, "two pair" and "three of a kind" are treated as the same category but not in the classical test. Likewise, "full house" and "four of a kind" are treated as the different category in the classical test.

In SPRNG, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.5 Coupon collector's test

We generate random integers in $[0, d-1]$ and observe the length of the segment that includes a complete set of integers from 0 to $d-1$. For example, if we get the following random integer sequence for $d = 3$,

$$0, 1, 1, 2, 0, 0, 0, 1, 0, 1, 0, 0, 2, 0, 1, 2, 0, 0, 1, 2, \cdots,$$

then we obtain the following result.

segment	length of segment
(0, 1, 1, 2)	4
(0, 0, 0, 1, 0, 1, 0, 0, 2)	9
(0, 1, 2)	3
(0, 0, 1, 2)	4
...	

Usually, we lump segments of length larger than t and have $t - d + 1$ categories. A chi-square test is then applied to these categories.

In SPRNG, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.6 Permutation test

We generate n sets of m successive RNs (m-tuples) in $[0, 1)$. The RNs in each set have $m!$ possible orders and the number of times each order appears is scored. All the orders must occur with equal probability if the RNs are properly generated. A chi-square test is thus applied to $m!$ categories with probability $1/m!$.

As an example, suppose that we get the following RN sequence,

0.10574, 0.66509, 0.46622, 0.93925, 0.26551, 0.11361,
0.25714, 0.45412, 0.13971, 0.59733, 0.26273, 0.09938,
$\cdots$,

and consider the sets of triples ($m = 3$). When we rank the triples in each set according to their magnitude, we have 6 categories; (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1), where 1 and 3 mean the smallest and largest RNs in each set, respectively. Then we can obtain the following result from the above sequence.

triples	category
(0.10574, 0.66509, 0.46622)	(1,3,2)
(0.93925, 0.26551, 0.11361)	(3,2,1)
(0.25714, 0.45412, 0.13971)	(2,3,1)
(0.59733, 0.26273, 0.09938)	(3,2,1)
...	

In SPRNG, the chi-square test is repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.7 Runs-up test

In the runs-up test, RNs are generated in $[0, 1)$ and the length of runs-up in which the successive RNs are increasing. For example, if we get the same RN sequence as in the permutation test and put a vertical line at the breakpoint,

$$
\begin{gathered}
0.10574,\ 0.66509 \mid 0.46622,\ 0.93925 \mid 0.26551 \mid 0.11361, \\
0.25714,\ 0.45412 \mid 0.13971,\ 0.59733 \mid 0.26273 \mid 0.09938, \\
\cdots,
\end{gathered}
$$

then the length of the first run is 2, the length of the second run is 2, the length of the third and fourth runs is 1, etc. The runs up of the length greater than t are lumped together.

We cannot simply apply a chi-square test to the counts of the length because the adjacent runs are not independent. Instead we apply the chi-square test to a test statistic in the covariance matrix form.

In SPRNG, a slightly modified version of the test is implemented. The RN that follows a previous run is discarded. In the above example, 0.46622, 0.26551, 0.13971 and 0.26273 are discarded;

$$
\begin{gathered}
0.10574,\ 0.66509 \mid (0.46622)\ 0.93925 \mid (0.26551) \mid 0.11361, \\
0.25714,\ 0.45412 \mid (0.13971)\ 0.59733 \mid (0.26273) \mid 0.09938, \\
\cdots.
\end{gathered}
$$

Then the lengths of runs-up are, in turn, 2, 1, 3, 1, 1 $\cdots$. The chi-square test is applied to the counts of the lengths and repeated the specified times (NTESTS) and the K-S test is applied for the obtained chi-square statistics.

3.2.8 Maximum-of-t test

We generate n sets of t successive RNs (t-tuples) in $[0, 1)$ and observe a maximum RN in each set. For example, suppose that we get the following RN sequence,

$$0.10574,\ 0.66509,\ 0.46622,\ 0.93925,\ 0.26551,\ 0.11361,$$
$$0.25714,\ 0.45412,\ 0.13971,\ 0.59733,\ 0.26273,\ 0.09938,$$
$$\cdots.$$

If $t = 3$, we obtain the following result.

triples	maximum RN
(0.10574, 0.66509, 0.46622)	0.66509
(0.93925, 0.26551, 0.11361)	0.93925
(0.25714, 0.45412, 0.13971)	0.45412
(0.59733, 0.26273, 0.09938)	0.59733
$\cdots$	

The distribution of the maximum RNs should be x^t and the K-S test is applied to them.

In SPRNG, the K-S test is repeated the specified times (NTESTS) and another K-S test is applied for the obtained K-S statistics.

3.2.9 Collision test

Suppose that we have m urns and throw n balls into the urns at random. If $m >> n$, then most of the balls fall into empty urns. However, some balls may fall into an run that is occupied by other balls. In this case, it is said that a "collision" has occurred. The collision test counts the number of collisions and a RNG passes this test if there are not too many or too few collisions.

In order to realize the above idea, we generate n sets of $\log md$ successive random integers in $[0, 2^{\log d} - 1]$. Then we form n new $\log m$ bit random integers with the $\log d$ most significant bits from $\log md$ random integers, where $\log m = \log md \times \log d$. For example, if $\log d = 1$ and we get the following random integer sequence,

$$0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,$$
$$0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,$$
$$1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 1,\ 1,\ 0,$$
$$\cdots,$$

then we obtain the following result for $\log md = 20$.

$$\begin{aligned} 01010000010010010001_2 &= 328849 \\ 00101010111101100010_2 &= 175970 \\ 11100001010100010110_2 &= 922902 \\ &\cdots. \end{aligned}$$

All possible values of the new random integers and each new random integer correspond to urns and a ball, respectively. When the same random integer appears in n sets, a collision occurs. The number of collisions is counted and a chi-square test is applied to it.

In SPRNG, $\log m = \log md \times \log d$ must be less than 32 and n must be less than the number of possible new random integers $2^{\log md \times \log d}$.

3.3 DIEHARD Tests

3.3.1 Birthday spacings test

In this test, we choose m birthdays in a year of n days. This is simulated by generating m random integers in $[1, n]$. Suppose we get random integers $I_1, I_2, \cdots, I_m$, we sort them into non-decreasing order; $I_{(1)} \leq I_{(2)} \leq \cdots \leq I_{(m)}$. Then we obtain a list of m birthday spacings;

$$I_{(1)}, I_{(2)} - I_{(1)}, I_{(3)} - I_{(2)}, \cdots, I_{(m)} - I_{(m-1)} = Y_1, Y_2, Y_3, \cdots, Y_m.$$

We sort the spacings into non-decreasing order; $Y_{(1)} \leq Y_{(2)} \leq \cdots \leq Y_{(m)}$. Then we counts the number of indices j such that $1 < j \leq n$ and $Y_{(j)} = Y_{(j-1)}$. If j is the number of values that occur more than once in that list, then j is asymptotically Poisson distributed with mean $m^3/(4n)$.

Experience shows n must be quite large, say $n \geq 2^{18}$, for comparing the results to the Poisson distribution with that mean. This test in DIEHARD uses $n = 2^{24}$ and $m = 2^9$, so that the underlying distribution for j is taken to be Poisson with mean $\lambda = (2^9)^3/(2^2 \times 2^{24}) = 2$. The process to obtain j is repeated 500 times and a chi-square test is applied to 500 j's. As a result, the chi-square test provides a p-value.

This test in DIEHARD uses several parts of bits of given 32-bit random integers. The first test uses bits 1-24 (counting from the left) from integers. In the second test, bits 2-25 are used to provide birthdays, then 3-26 and so on to bits 9-32. Each set of bits provides a p-value, and the nine p-values provide a sample for a K-S test.

3.3.2 Overlapping 5-permutation test

This test is a kind of overlapping m-tuple tests. The tests use sets of overlapped successive random integers. For example, we consider the following sequence of random integers obtained for $d = 8$ in Eq. (5);

$$0,\ 5,\ 3,\ 7,\ 2,\ 0,\ 2,\ 3,\ 1,\ 4,\ 2,\ 0,\ 7,\ 1,\ 2,\ 7,\ \cdots,\ 4,\ 5,\ 3,\ 1,\ 5,\ 2$$

In the case of $m = 5$, we add the first 4 integers to the end of the sequence and we group n sets of overlapping 5-tuples;

$$(0,\ 5,\ 3,\ 7,\ 2),\ (5,\ 3,\ 7,\ 2,\ 0),\ (3,\ 7,\ 2,\ 0,\ 2),\ \cdots\ (5,\ 2,\ 0,\ 5,\ 3),\ (2,\ 0,\ 5,\ 3,\ 7)$$

According to Marsaglia[24], the circulation has an asymptotically negligible effect but makes deriving a covariance matrix for a test statistic much simpler. Obviously, the sets are not independent of each other and thus a test statistic of the quadratic form with a covariance matrix is used. The statistic has asymptotically a chi-square distribution.

The basic idea of the overlapping 5-permutation test is the same as the permutation test described in Section 3.2.6. The difference is whether the sets of 5-tuple is overlapped or not. Each set of five successive integers can be in one of 120 states (5! possible orderings of five integers). The number of occurrences of each state is counted for the test statistic.

This test in DIEHARD uses random integer sequences of length 1000 and forms 1000 sets of overlapping 5-tuples. This process is repeated 1000 times and the cumulative counts are made for a million 32-bit random integers. The counts are used to yield the test statistic with the quadratic form in the weak inverse of the 120×120 covariance matrix. (If $CC^{-}C = C$, then C^{-} is a weak inverse of C.) Finally a p-value is obtained from a chi-square distribution with 99 degrees of freedom (the asymptotic rank of the covariance matrix). This version of overlapping 5-permutation test uses a million integers, twice.

3.3.3 Binary rank test

We form a binary matrix from a sequence of random integers. Each column of the matrix consists of the binary representation of a random integer. In general, m n-bit random integers forms a $m \times n$ binary matrix. The i-th n-bit random integer can be expressed as follows;

$$\begin{aligned} I_i &= f_{i,1} * 2^{n-1} + f_{i,2} * 2^{n-1} + \cdots + f_{i,n-1} * 2^1 + f_{i,n} * 2^0 \\ &= (f_{i,1} f_{i,2} \cdots f_{i,n-1} f_{i,n}), \end{aligned}$$

where $f_{i,j}$ is 0 or 1. Then using m integers, we obtain a binary matrix A;

$$A = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{pmatrix}.$$

A lot of matrices are usually generated from a sequence of random integers and the ranks of the matrices are calculated. A chi-square test is applied to the ranks to obtain a p-value.

It is not alway necessary to use a full matrix for this test and we can use a partial matrix. The binary rank test in DIEHARD is performed for three forms of matrices; 31×31, 32×32(full) and 6×8 matrices. For 31×31 matrices, the leftmost 31 bits of 31 random integers are used to form each matrix. The ranks can be from 0 to 31, but ranks less than 28 are rare. Thus the counts for rank less than 28 are lumped together. Ranks are found for 40,000 matrices and a chi-square test is applied to counts for ranks 31,30,29 and equal to or less than 28.

For 32×32 matrices, all bits of 32 random integers are used to form each matrix. The ranks can be from 0 to 32. Since ranks less than 29 are rare, the counts for rank less than 29 are lumped together. Ranks are found for 40,000 matrices and a chi-square test is applied to counts for ranks 32, 31, 30 and equal to or less than 29.

For 6×8 matrices, 6 bits of 8 random integers are used to form each matrix. The ranks can be from 0 to 6. However, ranks 0,1,2,3 are rare and thus their counts are lumped together as rank 4. Ranks are found for 100,000 matrices and a chi-square test is applied to the counts for ranks 6,5 and equal to or less than 4.

3.3.4 Bitstream test

In this test, a sequence of random integers is taken to be a stream of sequential bits. Since the i-th 32-bit random integer is expressed as $(b_{i,1}b_{i,2}\cdots b_{i,32})$ where $b_{i,j} = 0$ or 1, the stream becomes

$$b_{1,1}, b_{1,2}, \cdots, b_{1,32}, b_{2,1}, b_{2,2}, \cdots, b_{2,32}, \cdots, b_{i,1}, b_{i,2}, \cdots, b_{i,32}, \cdots.$$

We treat $b_{i,j}$'s as a letter 0 or 1 and think of the stream of bits as a succession of overlapping 20-letter "words". The first word is $b_{1,13}b_{1,14}\cdots b_{1,31}b_{1,32}$ and the second word is $b_{1,14}b_{1,15}\cdots b_{1,32}b_{2,1}$, and so on. The bitstream test counts the number of missing 20-letter (20-bit) words in a string of 2^{21} overlapping 20-letter words. There are 2^{20} possible 20 letter words. For a truly random string of $2^{21}+19$ bits, the number of missing words j should be (very close to) normally distributed with mean 141,909 and standard deviation $\sigma = 428$. Thus $(j-141909)/428$ should be a standard normal variate ($z = (x-\mu)/\sigma$) that leads to a uniform $[0,1)$ p-value. The test in DIEHARD is repeated twenty times.

3.3.5 Overlapping-pairs-sparse-occupancy test (OPSO test)

In this test, 2-letter words are formed from an alphabet of 1024 letters. Each letter is determined by a designated string of consecutive 10 bits from a 32-bit random integer in the sequence to be tested. When we express the i-th 32-bit random integer as $(b_{i,1}b_{i,2}\cdots b_{i,32})$ in the binary form, we can form 2-letter words with 2 last 10 bits;

$$\underbrace{b_{1,1}b_{1,2}\cdots b_{1,32}}_{\text{32-bit integer}},\underbrace{b_{2,1}b_{2,2}\cdots b_{2,32}}_{\text{32-bit integer}},\cdots$$

$$\Longrightarrow \overbrace{\underbrace{b_{1,13}b_{1,14}\cdots b_{1,32}}_{\text{1 letter}}\underbrace{b_{2,13}b_{2,14}\cdots b_{2,32}}_{\text{1 letter}}}^{\text{1 word}},\overbrace{\underbrace{b_{2,13}b_{2,14}\cdots b_{2,32}}_{\text{1 letter}}\underbrace{b_{3,13}b_{3,14}\cdots b_{3,32}}_{\text{1 letter}}}^{\text{1 word}},\cdots$$

The test generates 2^{21} overlapping 2-letter words (from $2^{21}+1$ "keystrokes") and counts the number of missing words, that is, 2-letter words which do not appear in the entire sequence. The number of missing words j should be very close to normally distributed with mean 141,909, standard deviation $\sigma = 290$. Thus $(j-141909)/290$ should be a standard normal variate that provide a p-value.

The above process is repeated for the next designated 10 bits of 32-bit random integers of the same sequence. In the next process, the following 2-letter words are used;

$$\overbrace{\underbrace{b_{1,12}b_{1,13}\cdots b_{1,31}}_{\text{1 letter}}\underbrace{b_{2,12}b_{2,13}\cdots b_{2,31}}_{\text{1 letter}}}^{\text{1 word}},\overbrace{\underbrace{b_{2,12}b_{2,13}\cdots b_{2,31}}_{\text{1 letter}}\underbrace{b_{3,12}b_{3,13}\cdots b_{3,31}}_{\text{1 letter}}}^{\text{1 word}},\cdots$$

The OPSO test in DIEHARD repeats the process 22 times with the designated 10 bits shifted left.

3.3.6 Overlapping-quadruples-sparse-occupancy test (OQSO test)

The OQSO test is similar to the OPSO test above. In this test, 4-letter words are formed from an alphabet of 32 letters. Each letter is determined by a designated string of 5 consecutive bits from a 32-bit random integer in the sequence to be tested. Using the same expression for the i-th 32-bit random integer as in the OPSO test, we can form 4-letter words with 2 last 5 bits;

$$\overbrace{\underbrace{b_{1,28}b_{1,29}\cdots b_{1,32}}_{\text{1 letter}}\underbrace{b_{2,28}b_{2,29}\cdots b_{2,32}}_{\text{1 letter}}\underbrace{b_{3,28}b_{3,29}\cdots b_{3,32}}_{\text{1 letter}}\underbrace{b_{4,28}b_{4,29}\cdots b_{4,32}}_{\text{1 letter}}}^{\text{1 word}},$$

$$\overbrace{\underbrace{b_{2,28}b_{2,29}\cdots b_{2,32}}_{\text{1 letter}}\underbrace{b_{3,28}b_{3,29}\cdots b_{3,32}}_{\text{1 letter}}\underbrace{b_{4,28}b_{4,29}\cdots b_{4,32}}_{\text{1 letter}}\underbrace{b_{5,28}b_{5,29}\cdots b_{5,32}}_{\text{1 letter}}}^{\text{1 word}},$$

$$\vdots$$

The test generates 2^{21} overlapping 4-letter words (from $2^{21}+3$ "keystrokes") and counts the number of missing words, that is, 4-letter words which do not appear in the entire sequence. The number of missing words j should be very close to normally distributed with mean 141909, standard deviation $\sigma = 295$. Thus $(j - 141909)/295$ should be a standard normal variate that provide a p-value.

The above process is repeated for the next designated 5 bits of 32-bit random integers of the same sequence. The OPSO test in DIEHARD repeats the process 28 times with the designated 10 bits shifted left.

3.3.7 DNA test

The DNA test is similar to the OPSO and OQSO tests above. In this test, 10-letter words are formed from an alphabet of 4 letters. Each letter is determined by a designated string of 2 consecutive bits from a 32-bit random integer in the sequence to be tested. Using the same expression for the i-th 32-bit random integer as in the OPSO test, we can form 10-letter words with 2 last 2 bits;

$$\overbrace{\underbrace{b_{1,31}b_{1,32}}_{\text{1 letter}}\underbrace{b_{2,31}b_{2,32}}_{\text{1 letter}}\cdots\underbrace{b_{10,31}b_{10,32}}_{\text{1 letter}}}^{\text{1 word}},\overbrace{\underbrace{b_{2,31}b_{2,32}}_{\text{1 letter}}\underbrace{b_{3,31}b_{3,32}}_{\text{1 letter}}\cdots\underbrace{b_{11,31}b_{11,32}}_{\text{1 letter}}}^{\text{1 word}},\cdots$$

The test generates 2^{21} overlapping 10-letter words (from $2^{21}+9$ "keystrokes") and counts the number of missing words, that is, 10-letter words which do not appear in the entire sequence. The number of missing words j should be very close to normally distributed with mean 141909, standard deviation $\sigma = 399$. Thus $(j - 141909)/295$ should be a standard normal variate that provide a p-value.

The above process is repeated for the next designated 2 bits of 32-bit random integers of the same sequence. The OPSO test in DIEHARD repeats the process 31 times with the designated 2 bits shifted left.

3.3.8 Count-the-1's test on a stream of bytes

This test is a kind of overlapping m-tuple tests. We consider a sequence of 32-bit random integers as a stream of bytes (4 bytes per 32 bit integer).

$$\overbrace{\underbrace{b_{1,1}\cdots b_{1,8}}_{\text{1 byte}},\underbrace{b_{1,9}\cdots b_{1,16}}_{\text{1 byte}},\underbrace{b_{1,17}\cdots b_{1,24}}_{\text{1 byte}},\underbrace{b_{1,25}\cdots b_{1,32}}_{\text{1 byte}}}^{\text{32-bit integer}},\underbrace{b_{2,1}\cdots b_{2,8}}_{\text{1 byte}},\cdots$$

Each byte can contain from 0 to 8 1's, with probabilities 1,8,28,56,70,56,28,8,1 over 256. Now let the stream of bytes provide a string of overlapping 5-letter words, each "letter" taking values A,B,C,D,E. The letters are determined by the number of 1's in a byte;

Number of 1's	Letter	Probability
0,1,2	A	37
3	B	56
4	C	70
5	D	56
6,7,8	E	37

There are 5^5 possible 5-letter words and the frequencies for each word are counted for a string of 2560000 overlapping 5-letter words.

The quadratic form in the weak inverse of the covariance matrix of the cell counts has asymptotically a chi-square distribution. Instead, an alternative statistic $Q_5 - Q_4$ is used to provide a p-value. Q_5 and Q_4 are the native Pearson's sums for the counts of 5- and 4- letter words, respectively, and defined as follows;

$$\begin{aligned} Q_5 &= \sum_{i,j,k,\ell,m} \frac{(w_{i,j,k,\ell,m} - \mu_{i,j,k,\ell,m})^2}{\mu_{i,j,k,\ell,m}} \\ Q_4 &= \sum_{i,j,k,\ell} \frac{(w_{i,j,k,\ell} - \mu_{i,j,k,\ell})^2}{\mu_{i,j,k,\ell}}, \end{aligned}$$

where w and μ are the observed and expected counts, respectively, and (i, j, k, ℓ, m) denotes a possible state (word). Then the statistic has asymptotically a chi-square distribution with $5^5 - 5^4$ degrees of freedom.

In DIEHARD, the above process is repeated twice and 2 p-values are obtained.

3.3.9 Count-the-1's test for specific bytes

This test is similar to the count-the-1's test on a stream of bytes. Again, we consider a sequence of 32-bit random integers as a stream of bytes. In this test, a specific byte in each integer is chosen to form a letter. For example, suppose the leftmost 8 bits in each integer are chosen, the following byte stream is obtained;

$$\underbrace{b_{1,1}\cdots b_{1,8}}_{\text{1 byte}}, \underbrace{b_{2,1}\cdots b_{2,8}}_{\text{1 byte}}, \underbrace{b_{3,1}\cdots b_{3,8}}_{\text{1 byte}}, \underbrace{b_{4,1}\cdots b_{4,8}}_{\text{1 byte}}, \underbrace{b_{5,1}\cdots b_{5,8}}_{\text{1 byte}}, \cdots.$$

¿From the stream, 256000 overlapping 5-letter words are formed and a test statistic to provide a p-value is calculated in the same way as the count-the-1's test on a stream of bytes.

Next, the process is performed for another byte stream comprised of a next specific byte in each integer,

$$\underbrace{b_{1,2}\cdots b_{1,9}}_{\text{1 byte}}, \underbrace{b_{2,2}\cdots b_{2,9}}_{\text{1 byte}}, \underbrace{b_{3,2}\cdots b_{3,9}}_{\text{1 byte}}, \underbrace{b_{4,2}\cdots b_{4,9}}_{\text{1 byte}}, \underbrace{b_{5,2}\cdots b_{5,9}}_{\text{1 byte}}, \cdots.$$

The process is repeated 25 times and thus all possible successive bytes in each integer are considered.

3.3.10 Parking lot test

We consider parking cars randomly in a square of side 100. Each car occupies space of a circle of radius 1 there[1]. When cars are parked repeatedly, an

[1] It seems that a car occupies a square of side 1 in the DIEHARD program.

attempt to park a car may cause a crash with one already parked. Then the attempt is tried again at a new random location. Each attempt leads to either a crash or a success. If a car is successfully parked, the position of the car is added to the list of cars already parked. The number of cars successfully parked k is counted for a large number of attempts and a p-value is provided from the distribution determined by simulation.

This test in DIEHARD is performed for 12000 attempts. Simulation shows that k should have a very close normal distribution with mean 3523 and standard deviation 21.9 for those attempts. Thus $(k - 3523)/21.9$ should be a standard normal variable that provides a p-value. This process is repeated 10 times and a K-S test is applied to a sample of 10 p-values.

3.3.11 Minimum distance test

In this test, $n = 8000$ random points in a square of side 10000 are chosen and the minimum distance d between the $(n^2 - n)/2$ pairs of the points is scored. If the points are truly independent and uniform, the square of the minimum distance d^2 should be (very close to) exponentially distributed with mean 0.995. Thus $1 - \exp(-d^2/0.995)$ should be uniform on [0,1). This process is repeated 100 times. A K-S test on the resulting 100 values serves as a test of uniformity for random points in the square and yields a p-value.

3.3.12 3-D spheres test

In this test, 4000 random points are chosen in a cube of edge 1000. At each point, a sphere is centered large enough to reach the next closest point. Then the volume of the smallest such sphere is (very close to) exponentially distributed with mean $120\pi/3$. Thus the radius cubed r^3 is exponential with mean 30.0 (The mean is obtained by extensive simulation). The 3D spheres test in DIEHARD generates 4000 such spheres 20 times. Each minimum radius cubed leads to a uniform variable by means of $1 - \exp(-r^3/30.0)$, then a K-S test is performed on the 20 p-values.

3.3.13 Squeeze test

This test uses real-valued random numbers uniformly distributed on $[0, 1)$. The random numbers are generated from a sequence of 32-bit random integers as follows;

$$U_i = I_i/2^{32}.$$

An initial number $k_0 = 2^{31} = 2147483647$ is multiplied by a random number and then the next number k_1 is obtained with the following equation;

$$k_i = \lceil k_{i-1} U \rceil,$$

where $\lceil x \rceil$ is the ceiling of x, that is, the least integer such that $\min_{k \geq x} k$. The reduction is repeated until k_j is 1 and j is the number of iterations necessary to reduce k to 1. In DIEHARD, 100000 j's are found and then the number of times that j is $\leq 6, 7, \cdots, 47, \geq 48$ is counted. A chi-square test is applied to the counts to provide a p-value.

3.3.14 Overlapping sums test

This test also uses real-valued random numbers uniformly distributed on $[0, 1)$ and the numbers are obtained in the same way as the squeeze test. Suppose we get a sequence of the random numbers,

$$U_1, U_2, \cdots,$$

then we can form overlapping sums of 100 random numbers;

$$S_1 = U_1 + \cdots + U_{100}, S_2 = U_2 + \cdots + U_{101}, \cdots.$$

The S's are virtually normal with a certain covariance matrix. A linear transformation of the S's yields a sequence of independent standard normals, which are converted to uniform variables for a K-S test. This process is repeated 100 times and 100 p-values are obtained. Another K-S test is performed on the 100 p-values to provide a final p-value. Furthermore, the above process is repeated 10 times in DIEHARD.

3.3.15 Runs test

This is basically the same as the runs-up test in the standard test suite but this test in DIEHARD includes the runs-down test. The test counts runs-up and runs-down in a sequence of real-valued random numbers uniformly distributed on $[0, 1)$. The numbers are obtained from 32-bit integers in the same way as the squeeze test.

The covariance matrices for the runs-up and runs-down are well known, leading to chi-square tests for quadratic forms in the weak inverses of the covariance matrices. The runs are counted for sequences of length 10,000 and this is repeated 10 times to yield a p-value. Furthermore, this process is repeated twice.

3.3.16 Craps test

This test simulates the game of craps where a player always makes a "pass-line" bet. The craps game is based on the rolls of 2 dice. For the first throw of the dice ("come-out roll"), the player wins the pass-line bet if the come-out roll is either a 7 or 11. The player loses the pass-line bet if the come-out roll is a 2, 3 or 12 (Craps). If the come-out roll is any other than the above (4, 5, 6, 8, 9, 10), the roll is set to a "point" and the game continues. For the second throw or later, the player wins if the point appears again before a 7 is rolled. The player loses if a 7 is rolled before the point appears again.

Each 32-bit random integer I provides the value for the throw of a die with $(I/2^{32}) \times 6 + 1$. The test in DIEHARD plays 200000 games of craps and counts the number of wins and the number of throws necessary to end each game. The number of wins j should be (very close to) a normal with mean $\mu = 200000p$ and variance $\sigma^2 = 200000p(1-p)$ with $p = 244/495$. Thus $(j-\mu)/\sigma$ should be a standard normal variate that yields a p-value.

The number of throws necessary to complete the game can vary from 1 to infinity, but counts for all larger than 21 are lumped with 21. A chi-square test is performed on the counts for the number of throws to provide a p-value.

4 Test Results

4.1 Results for the spectral test

In order to perform the spectral test, we employed an algorithm proposed by Hopkins [8]. We transformed a provided source code written in Fortran 66 into a script bc that is an arbitrary precision numeric processing language supported by Free Software Foundation [14]. With the bc script, we obtained the measures $\mu_t(m, g)$, $S_t(m, g)$ and $M_T(m, g)$.

At first, we obtained the measures for LCG$(69069, 0, 2^{32})$ and LCG(69069, 1, 2^{32}) to verify that the transformed script works correctly. These RNGs are proposed by Marsaglia [15] and the values of $\mu_t(m, g)$ and $S_t(m, g)$ are listed in literatures [3, p. 107] and [16, p. 616]. Tables 2 and 3 show the results of the spectral test for the above LCGs. Our results are in very good agreement with Fishman's and Knuth's ones. Therefore, it has been verified that the transformed script gives correct values.

Table 2: Results of the spectral test for LCG$(69069, 0, 2^{32})$

	Our results		Fishman[16, p. 616]
Dimension (t)	$\mu_t(69069, 0, 2^{32})$	$S_t(69069, 0, 2^{32})$	$S_t(69069, 0, 2^{32})$
2	0.7759	0.4625	0.4625
3	0.1819	0.3131	0.3131
4	0.4312	0.4572	0.4572
5	0.7694	0.5529	0.5529
6	0.0682	0.3767	0.3767

Table 3: Results of the spectral test for LCG$(69069, 1, 2^{32})$

	Our results		Knuth[3, p. 107]
Dimension (t)	$\mu_t(69069, 1, 2^{32})$	$S_t(69069, 1, 2^{32})$	$\mu_t(69069, 1, 2^{32})$
2	3.1037	0.9250	3.10
3	2.9099	0.7890	2.91
4	3.2036	0.7548	3.20
5	5.0065	0.8042	5.01
6	0.0171	0.2990	0.02

Table 4 shows the results of the spectral test for the current MCNP RNG and LCGs proposed as new MCNP RNGs. The μ_t values less than 0.1 are bold-faced. According to Knuth's criterion, the MCNP RNG pass the spectral test but the extended LCGs (LCG 2 $\sim$ 7) fail. This indicates that simple extension from the original MCNP RNG to 63-LCGs are not good.

On the other hand, other 63-bit LCGs proposed by L'Ecuyer, of course, pass the test with excellent μ_t or S_t values because their multipliers are chosen based on this test. Our M_8 values coincide with the values in L'Ecuyer's paper [2]. It also ensures that our program calculates correct results of the spectral test.

Table 4: Results of the spectral test for LCGs proposed as new MCNP RNGs

Dimension (t)	2	3	4	5	6	7	8
LCG($5^{19}, 0, 2^{48}$)							
μ_t	3.0233	0.1970	1.8870	0.9483	1.8597	0.8802	1.2931
S_t	0.9129	0.3216	0.6613	0.5765	0.6535	0.5844	0.6129
LCG($5^{19}, 0, 2^{63}$)							
μ_t	1.7321	2.1068	2.7781	1.4379	**0.0825**	2.0043	5.9276
S_t	0.6910	0.7085	0.7284	0.6266	0.3888	0.6573	0.7414
LCG($5^{23}, 0, 2^{63}$)							
μ_t	**0.0028**	1.9145	2.4655	5.4858	0.3327	0.2895	6.6286
S_t	0.0280	0.6863	0.7070	0.8190	0.4906	0.4986	0.7518
LCG($5^{25}, 0, 2^{63}$)							
μ_t	0.3206	1.8083	**0.0450**	3.0128	0.3270	3.1053	0.4400
S_t	0.2973	0.6733	0.2598	0.7265	0.4892	0.6998	0.5356
LCG($5^{19}, 1, 2^{63}$)							
μ_t	1.7321	2.9253	2.4193	0.3595	**0.0206**	0.5011	1.6439
S_t	0.6910	0.7904	0.7036	0.4749	0.3086	0.5392	0.6316
LCG($5^{23}, 1, 2^{63}$)							
μ_t	**0.0007**	2.8511	2.5256	3.1271	4.5931	1.8131	4.2919
S_t	0.0140	0.7837	0.7112	0.7319	0.7598	0.6480	0.7121
LCG($5^{25}, 1, 2^{63}$)							
μ_t	**0.0801**	3.4624	1.3077	1.0853	1.4452	0.7763	1.3524
S_t	0.1486	0.8361	0.6033	0.5923	0.6266	0.5740	0.6163
LCG($3512401965023503517, 0, 2^{63}$)							
μ_t	2.9062	2.9016	3.1105	4.0325	5.3992	6.7498	7.2874
S_t	0.8951	0.7883	0.7493	0.7701	0.7806	0.7818	0.7608
LCG($2444805353187672469, 0, 2^{63}$)							
μ_t	2.2588	2.4430	6.4021	2.9364	3.0414	5.4274	4.6180
S_t	0.7891	0.7443	0.8974	0.7228	0.7094	0.7579	0.7186
LCG($1987591058829310733, 0, 2^{63}$)							
μ_t	2.4898	3.4724	1.7071	2.5687	2.1243	2.0222	4.1014
S_t	0.8285	0.8369	0.6449	0.7037	0.6682	0.6582	0.7080
LCG($9219741426499971445, 1, 2^{63}$)							
μ_t	2.8509	2.8046	3.5726	3.8380	3.8295	6.4241	6.8114
S_t	0.8865	0.7794	0.7757	0.7625	0.7371	0.7763	0.7544
LCG($2806196910506780709, 1, 2^{63}$)							
μ_t	1.9599	4.0204	4.4591	3.1152	3.0728	3.0111	3.7947
S_t	0.7350	0.8788	0.8199	0.7314	0.7106	0.6967	0.7012
LCG($3249286849523012805, 1, 2^{63}$)							
μ_t	2.4594	2.4281	3.7081	2.8333	3.7633	3.0844	1.9471
S_t	0.8234	0.7428	0.7829	0.7176	0.7350	0.6991	0.6451

4.2 Results for standard test suite

All tests calculate the values of a test statistic and they are evaluated with chi-square or K-S goodness-of-fit tests. As described in Section 3.2, all the standard tests except for the collision test in SPRNG includes two steps; the first step is a chi-square or K-S test for subsequences and the second step is a K-S test for the resultant percentiles in the first step. This procedure is called a second-order test [18] or a two-level test [19] and may tend to detect both local and global nonrandomness of a random number sequence [3, p. 52]. The collision test in SPRNG is a first-order or single-level test.

The goodness-of-fit test yields a p-value defined by

$$p = F(t) = \Pr(T < t) \tag{21}$$

where $F(t)$ is a distribution function for a value t of a test statistic T and T is a random variable. The p-value means that a test statistic is less than t with probability p. For the chi-square and K-S tests, $F(t)$ is the chi-square distribution and a distribution derived by Birnbaum [20], respectively. The approximated form of the distribution is often used for the K-S test [21] and the SPRNG test routines use this form.

RNGs are evaluated by the p-value. A RNG fails a test if a p-value of the test is close to 0 or 1. Otherwise, the RNG passes the test. The most difficult problem for the evaluation is to determine a significance level. The level is usually 0.05 or 0.01 which is based on experiences. In this work, we set the significance level to 0.01 and perform each test 3 times for disjoint random number sequences. We consider that a RNG fails only if all 3 p-values are less than 0.01 (1%) or larger than 0.99 (99%).

One requires some parameters for the standard tests since the default values are not provided for them in SPRNG. We have chosen them from papers where some parameters are listed. The parameters used are L'Ecuyer's[13] and Vattulainen's set[22] listed in Table 5 and 6, respectively.

Using these parameters, we performed the standard tests for all 13 RNGs in the new MCNP random package. Each test was repeated 3 times for 3 disjoint random number sequences. To ensure the sequences are disjoint, an initial seed for each sequence is set to the final value of the previous sequence. Namely, we used 3 consecutive sequences.

Tables 7 $\sim$ 19 show the results of the standard tests for 13 RNGs. Suspicious p-values that are less than 0.01 (1%) or larger than 0.99 (99%) are bold-faced. All the RNGs pass all the tests for L'Ecuyer's and Vattulainen's test suites.

Table 5: Parameters for L'Ecuyer's test suite

Standard tests	Parameters	Test ID
Equidistribution	$N = 10^4, n = 10^3, d = 64$	LEC01
	$N = 10^4, n = 10^4, d = 256$	LEC02
Serial	$N = 10^3, n = 10^5, d = 64$	LEC03
Gap	$N = 10^3, n = 10^4, a = 0.0, b = 0.05, t = 15$	LEC04
	$N = 10^3, n = 10^4, a = 0.95, b = 1.0, t = 15$	LEC05
	$N = 10^3, n = 10^4, a = 1/3, b = 2/3, t = 10$	LEC06
Poker	$N = 10^3, n = 10^4, k = 4, d = 4$	LEC07
	$N = 10^3, n = 10^4, k = 6, d = 8$	LEC08
	$N = 10^3, n = 10^4, k = 8, d = 16$	LEC09
Coupon	$N = 10^3, n = 10^4, d = 5, t = 25$	LEC10
Permutation	$N = 10^3, n = 10^4, t = 3$	LEC11
	$N = 10^3, n = 10^4, t = 5$	LEC12
Runs-up	$N = 10^3, n = 10^5, t = 6^*$	LEC13
Maximum of t	$N = 10^3, n = 10^4, t = 8$	LEC14
Collision	$N = 10^2, n = 2 \times 10^4, \log md = 6, \log d = 3$	LEC15
	$N = 10^2, n = 2 \times 10^4, \log md = 10, \log d = 2$	LEC16
	$N = 10^2, n = 2 \times 10^4, \log md = 20, \log d = 1$	LEC17

N is the number of times the test was repeated for the (second-level) K-S test. n is the length of the random number sequence. Other parameters are described in Section 3.2.
$*$) t is not listed in the paper[13], so it is set to the same value as Vattulainen's value for the runs-up test.

Table 6: Parameters for Vattulainen's test suite

Standard tests	Parameters	Test ID
Equidistribution	$N = 10^4, n = 10^4, d = 128$	VAT01
	$N = 10^4, n = 10^5, d = 256$	VAT02
Serial	$N = 10^3, n = 10^5, d = 100$	VAT03
Gap	$N = 10^3, n = 2.5 \times 10^4, a = 0.0, b = 0.05, t = 30$	VAT04
	$N = 10^3, n = 2.5 \times 10^4, a = 0.45, b = 0.55, t = 30$	VAT05
	$N = 10^3, n = 2.5 \times 10^4, a = 0.95, b = 1.0, t = 30$	VAT06
Runs-up*	$N = 10^3, n = 10^5, t = 6$	VAT07
Maximum of t	$N = 10^3, n = 2 \times 10^3, t = 5$	VAT08
	$N = 10^3, n = 2 \times 10^3, t = 3$	VAT08
Collision	$N = 10^3, n = 2^{14}, \log md = 2, \log d = 10$	VAT10
	$N = 10^3, n = 2^{14}, \log md = 4, \log d = 5$	VAT11
	$N = 10^3, n = 2^{14}, \log md = 10, \log d = 2$	VAT12

N is the number of times the test was repeated for the (second-level) K-S test. n is the length of the random number sequence. Other parameters are described in Section 3.2.
$*$) Same as L'Ecuyer's runs-up test.

Table 7: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(5^{19}, 0, 2^{48})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	49.94	75.04	**99.38**
	LEC02	54.36	84.90	40.75
Serial	LEC03	90.24	80.66	38.01
Gap	LEC04	56.79	61.72	18.74
	LEC05	49.91	2.09	24.06
	LEC06	89.51	65.11	87.18
Poker	LEC07	59.01	21.28	38.66
	LEC08	7.51	95.66	26.90
	LEC09	11.25	92.17	85.69
Coupon	LEC10	40.25	97.48	28.11
Permutation	LEC11	22.26	52.56	54.19
	LEC12	67.54	66.14	61.76
Runs-up	LEC13	49.87	39.21	92.83
Maximum of t	LEC14	52.26	37.63	87.46
Collision	LEC15	95.61	61.32	96.24
	LEC16	8.00	95.67	93.13
	LEC17	9.33	72.21	73.29
Vattulainen's test suite				
Equidistribution	VAT01	64.71	16.00	69.64
	VAT02	42.17	43.39	48.46
Serial	VAT03	31.45	93.43	88.68
Gap	VAT04	1.43	27.75	78.76
	VAT05	55.15	83.40	34.84
	VAT06	11.12	45.22	1.45
Runs-up	VAT07	49.87	39.21	92.83
Maximum of t	VAT08	39.03	66.30	41.71
	VAT09	81.50	46.55	77.76
Collision	VAT10	49.21	21.66	78.34
	VAT11	27.68	63.79	11.94
	VAT12	90.80	48.09	51.65

Table 8: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(5^{19}, 0, 2^{63})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	95.61	49.15	43.86
	LEC02	7.68	50.20	74.52
Serial	LEC03	18.03	97.98	11.65
Gap	LEC04	37.73	71.36	79.00
	LEC05	68.33	30.14	45.35
	LEC06	45.81	48.95	91.66
Poker	LEC07	60.72	28.14	30.19
	LEC08	33.67	69.57	96.30
	LEC09	57.81	81.96	7.30
Coupon	LEC10	58.37	**99.64**	40.32
Permutation	LEC11	91.65	52.83	67.19
	LEC12	1.24	49.35	14.86
Runs-up	LEC13	61.42	11.97	85.93
Maximum of t	LEC14	32.73	89.29	94.39
Collision	LEC15	12.57	27.34	29.43
	LEC16	92.09	54.02	51.15
	LEC17	91.57	16.30	36.57
Vattulainen's test suite				
Equidistribution	VAT01	26.06	4.13	94.13
	VAT02	49.22	28.22	83.85
Serial	VAT03	83.31	36.07	90.10
Gap	VAT04	70.22	82.45	49.52
	VAT05	88.86	69.45	47.13
	VAT06	59.50	8.70	36.74
Runs-up	VAT07	61.42	11.97	85.93
Maximum of t	VAT08	47.35	**0.11**	34.25
	VAT09	80.81	10.19	10.96
Collision	VAT10	9.48	90.53	36.32
	VAT11	18.96	24.84	13.26
	VAT12	78.94	87.87	14.92

Table 9: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(5^{23}, 0, 2^{63})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	26.96	90.63	82.37
	LEC02	24.63	87.94	**99.31**
Serial	LEC03	22.01	71.44	32.92
Gap	LEC04	89.94	19.32	6.98
	LEC05	97.79	89.05	14.95
	LEC06	90.78	31.90	14.66
Poker	LEC07	43.79	13.93	14.15
	LEC08	81.53	4.70	77.55
	LEC09	73.58	67.87	54.33
Coupon	LEC10	98.91	97.38	47.62
Permutation	LEC11	10.24	27.34	14.11
	LEC12	78.32	81.47	95.96
Runs-up	LEC13	44.39	18.39	66.05
Maximum of t	LEC14	73.77	59.14	16.98
Collision	LEC15	35.46	43.76	67.37
	LEC16	8.83	50.78	24.68
	LEC17	25.52	61.10	72.94
Vattulainen's test suite				
Equidistribution	VAT01	23.04	68.04	**99.31**
	VAT02	19.89	74.40	32.44
Serial	VAT03	95.96	66.15	49.78
Gap	VAT04	60.42	77.52	56.76
	VAT05	14.99	53.08	5.36
	VAT06	70.86	11.22	3.68
Runs-up	VAT07	44.39	18.39	66.05
Maximum of t	VAT08	18.46	78.19	59.45
	VAT09	46.39	17.90	40.59
Collision	VAT10	72.54	64.95	23.75
	VAT11	8.24	11.02	2.43
	VAT12	72.51	66.78	50.87

Table 10: Results of L'Ecuyer's and Vattulainen's test suites for LCG($5^{25}, 0, 2^{63}$)

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	79.90	93.18	91.06
	LEC02	45.11	95.23	47.81
Serial	LEC03	67.51	41.70	47.44
Gap	LEC04	79.52	99.50	35.82
	LEC05	60.67	39.82	17.22
	LEC06	81.25	35.42	79.54
Poker	LEC07	92.15	22.99	41.65
	LEC08	59.97	76.01	85.39
	LEC09	37.14	71.88	56.06
Coupon	LEC10	3.35	25.23	30.14
Permutation	LEC11	94.35	15.26	53.83
	LEC12	23.50	21.08	58.38
Runs-up	LEC13	47.01	72.52	71.53
Maximum of t	LEC14	41.59	23.38	69.78
Collision	LEC15	96.42	8.60	3.49
	LEC16	75.87	47.61	93.83
	LEC17	55.07	62.55	89.67
Vattulainen's test suite				
Equidistribution	VAT01	50.55	80.78	70.03
	VAT02	70.72	88.85	17.46
Serial	VAT03	83.63	54.71	72.20
Gap	VAT04	46.24	64.44	46.54
	VAT05	39.12	54.10	74.76
	VAT06	18.02	6.66	19.82
Runs-up	VAT07	47.01	72.52	71.53
Maximum of t	VAT08	37.92	54.86	24.81
	VAT09	9.19	16.34	2.86
Collision	VAT10	65.12	79.31	54.81
	VAT11	34.12	42.18	89.77
	VAT12	76.90	27.58	23.83

Table 11: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(5^{19}, 1, 2^{63})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	37.75	98.47	97.25
	LEC02	2.20	15.85	9.76
Serial	LEC03	85.94	77.91	34.27
Gap	LEC04	74.35	40.43	23.34
	LEC05	65.00	3.31	94.58
	LEC06	10.57	4.85	36.63
Poker	LEC07	15.82	10.03	76.45
	LEC08	32.75	34.97	9.39
	LEC09	2.26	90.75	81.20
Coupon	LEC10	34.13	28.71	64.86
Permutation	LEC11	75.58	93.36	90.57
	LEC12	83.84	38.55	92.90
Runs-up	LEC13	85.70	64.07	75.10
Maximum of t	LEC14	63.92	70.40	34.82
Collision	LEC15	18.13	77.26	26.97
	LEC16	65.52	11.54	12.91
	LEC17	16.14	33.95	50.35
Vattulainen's test suite				
Equidistribution	VAT01	42.92	98.81	48.52
	VAT02	30.77	29.72	88.60
Serial	VAT03	98.25	69.72	**0.83**
Gap	VAT04	59.80	57.33	50.33
	VAT05	53.91	61.56	63.91
	VAT06	37.34	81.74	40.55
Runs-up	VAT07	85.70	64.07	75.10
Maximum of t	VAT08	30.25	80.76	27.23
	VAT09	47.69	7.43	59.61
Collision	VAT10	5.95	75.31	72.28
	VAT11	83.64	84.87	7.94
	VAT12	54.09	58.00	8.29

Table 12: Results of L'Ecuyer's and Vattulainen's test suites for LCG($5^{23}, 1, 2^{63}$)

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	33.78	95.14	89.04
	LEC02	76.59	44.33	86.22
Serial	LEC03	77.63	10.18	34.20
Gap	LEC04	36.10	77.62	87.70
	LEC05	31.16	29.40	48.42
	LEC06	90.97	27.42	49.18
Poker	LEC07	62.23	40.58	72.69
	LEC08	64.77	89.30	11.11
	LEC09	72.97	75.33	87.47
Coupon	LEC10	23.73	65.07	88.32
Permutation	LEC11	68.21	32.47	21.60
	LEC12	86.50	88.58	92.04
Runs-up	LEC13	17.84	6.17	68.51
Maximum of t	LEC14	14.21	95.66	68.62
Collision	LEC15	2.82	19.73	98.52
	LEC16	71.06	31.75	52.53
	LEC17	83.93	27.00	64.96
Vattulainen's test suite				
Equidistribution	VAT01	41.97	72.84	35.51
	VAT02	82.31	37.91	41.86
Serial	VAT03	86.87	11.50	87.55
Gap	VAT04	43.40	93.39	19.63
	VAT05	87.92	53.51	65.02
	VAT06	65.55	42.36	**0.99**
Runs-up	VAT07	17.84	6.17	68.51
Maximum of t	VAT08	**0.71**	1.67	12.30
	VAT09	23.83	80.75	63.27
Collision	VAT10	61.06	89.98	68.18
	VAT11	45.48	47.67	9.98
	VAT12	11.58	22.94	97.77

Table 13: Results of L'Ecuyer's and Vattulainen's test suites for LCG($5^{25}, 1, 2^{63}$)

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	**99.69**	62.21	92.75
	LEC02	9.07	54.40	51.48
Serial	LEC03	37.41	44.02	85.73
Gap	LEC04	34.00	80.48	0.76
	LEC05	53.83	21.94	55.44
	LEC06	20.15	81.59	24.71
Poker	LEC07	55.38	7.63	11.06
	LEC08	40.00	15.39	4.67
	LEC09	54.16	7.28	54.47
Coupon	LEC10	52.43	30.01	29.40
Permutation	LEC11	47.82	62.82	38.59
	LEC12	69.91	5.07	95.52
Runs-up	LEC13	35.05	83.26	8.75
Maximum of t	LEC14	82.23	58.21	40.34
Collision	LEC15	97.12	95.28	20.24
	LEC16	29.03	42.35	7.94
	LEC17	21.37	34.13	25.30
Vattulainen's test suite				
Equidistribution	VAT01	18.14	88.64	48.88
	VAT02	3.61	62.97	81.79
Serial	VAT03	35.25	31.10	95.36
Gap	VAT04	73.46	3.09	59.98
	VAT05	60.76	62.98	80.49
	VAT06	79.11	97.23	30.52
Runs-up	VAT07	35.05	83.26	8.75
Maximum of t	VAT08	45.03	46.19	60.64
	VAT09	50.68	**0.55**	64.95
Collision	VAT10	41.02	62.24	75.09
	VAT11	36.51	78.98	84.25
	VAT12	51.07	18.92	40.06

Table 14: Results of L'Ecuyer's and Vattulainen's test suites for LCG($3512401965023503517, 0, 2^{63}$)

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	78.94	95.74	77.90
	LEC02	78.81	24.96	47.98
Serial	LEC03	3.97	10.42	92.03
Gap	LEC04	7.11	69.07	93.96
	LEC05	57.08	77.35	59.15
	LEC06	35.98	53.18	10.07
Poker	LEC07	84.66	19.67	41.14
	LEC08	62.51	23.18	71.31
	LEC09	73.33	7.01	76.54
Coupon	LEC10	38.70	6.32	49.40
Permutation	LEC11	31.19	58.89	**99.06**
	LEC12	53.44	83.87	71.22
Runs-up	LEC13	41.22	10.90	59.35
Maximum of t	LEC14	50.85	20.80	10.02
Collision	LEC15	29.85	28.54	17.82
	LEC16	27.34	12.05	80.14
	LEC17	65.85	76.39	2.44
Vattulainen's test suite				
Equidistribution	VAT01	44.03	60.90	63.39
	VAT02	51.33	86.86	14.12
Serial	VAT03	37.72	91.31	63.58
Gap	VAT04	58.42	4.11	44.37
	VAT05	43.06	35.81	78.08
	VAT06	92.01	67.67	80.22
Runs-up	VAT07	41.22	10.90	59.35
Maximum of t	VAT08	92.83	41.62	54.79
	VAT09	43.62	6.01	95.66
Collision	VAT10	46.00	68.38	56.47
	VAT11	70.06	65.61	40.86
	VAT12	86.35	34.77	48.93

Table 15: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(2444805353187672469, 0, 2^{63})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	95.14	79.03	71.62
	LEC02	76.57	37.08	10.07
Serial	LEC03	80.74	85.03	89.33
Gap	LEC04	14.41	60.21	8.88
	LEC05	7.49	46.79	2.62
	LEC06	59.45	28.83	28.25
Poker	LEC07	2.92	66.94	61.14
	LEC08	67.24	25.50	28.00
	LEC09	2.00	8.47	32.35
Coupon	LEC10	17.68	8.84	9.87
Permutation	LEC11	53.91	88.51	47.69
	LEC12	37.03	14.60	49.62
Runs-up	LEC13	81.47	26.66	24.05
Maximum of t	LEC14	84.26	0.89	10.17
Collision	LEC15	32.26	71.71	4.81
	LEC16	22.48	91.85	13.00
	LEC17	58.05	69.64	55.21
Vattulainen's test suite				
Equidistribution	VAT01	68.47	18.68	9.81
	VAT02	43.67	91.88	80.48
Serial	VAT03	54.33	78.96	69.55
Gap	VAT04	75.15	15.01	36.87
	VAT05	52.24	49.39	83.96
	VAT06	24.72	83.97	91.25
Runs-up	VAT07	81.47	26.66	24.05
Maximum of t	VAT08	60.06	35.55	12.10
	VAT09	40.52	32.16	34.65
Collision	VAT10	9.84	4.69	69.31
	VAT11	15.13	95.90	15.43
	VAT12	66.96	12.66	49.03

Table 16: Results of L'Ecuyer's and Vattulainen's test suites for LCG($1987591058829310733, 0, 2^{63}$)

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	93.63	98.18	83.34
	LEC02	44.22	32.07	64.11
Serial	LEC03	16.66	1.69	87.15
Gap	LEC04	30.77	52.82	92.52
	LEC05	56.67	85.33	74.06
	LEC06	31.34	**99.14**	95.24
Poker	LEC07	85.58	48.08	61.77
	LEC08	71.88	74.70	18.28
	LEC09	20.95	9.82	95.10
Coupon	LEC10	52.27	29.82	30.59
Permutation	LEC11	55.43	36.28	71.81
	LEC12	44.47	52.15	0.81
Runs-up	LEC13	68.33	38.44	49.67
Maximum of t	LEC14	6.50	58.20	10.07
Collision	LEC15	58.59	7.98	13.35
	LEC16	52.59	61.64	39.02
	LEC17	81.87	32.24	35.01
Vattulainen's test suite				
Equidistribution	VAT01	7.50	11.49	63.39
	VAT02	53.28	83.74	16.81
Serial	VAT03	95.53	13.08	49.88
Gap	VAT04	33.58	2.35	23.19
	VAT05	36.62	34.77	6.54
	VAT06	98.46	73.44	72.81
Runs-up	VAT07	68.33	38.44	49.67
Maximum of t	VAT08	**0.01**	2.42	94.93
	VAT09	33.16	59.16	**0.12**
Collision	VAT10	82.80	73.07	65.38
	VAT11	5.03	94.98	79.47
	VAT12	75.33	17.44	87.06

Table 17: Results of L'Ecuyer's and Vattulainen's test suites for LCG(9219741426499971445, 1, 2^{63})

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	24.85	78.67	82.55
	LEC02	42.85	77.22	57.85
Serial	LEC03	55.38	20.50	11.79
Gap	LEC04	41.76	45.09	29.37
	LEC05	49.80	13.52	69.07
	LEC06	39.53	53.32	65.63
Poker	LEC07	39.73	82.36	83.06
	LEC08	52.00	56.05	2.84
	LEC09	15.92	62.70	92.91
Coupon	LEC10	19.51	74.37	80.85
Permutation	LEC11	54.63	19.24	61.58
	LEC12	71.54	88.22	41.67
Runs-up	LEC13	64.28	**99.15**	39.88
Maximum of t	LEC14	75.10	89.41	41.23
Collision	LEC15	91.19	72.12	39.08
	LEC16	19.48	33.83	10.69
	LEC17	12.28	19.34	6.48
Vattulainen's test suite				
Equidistribution	VAT01	80.62	19.91	0.41
	VAT02	43.21	29.23	18.75
Serial	VAT03	17.29	21.21	59.01
Gap	VAT04	60.03	85.39	27.12
	VAT05	64.68	8.28	85.92
	VAT06	93.09	12.58	94.04
Runs-up	VAT07	64.28	**99.15**	39.88
Maximum of t	VAT08	37.01	30.52	31.36
	VAT09	63.52	4.24	49.61
Collision	VAT10	57.44	47.03	95.07
	VAT11	48.85	29.73	10.39
	VAT12	46.97	69.50	**99.29**

Table 18: Results of L'Ecuyer's and Vattulainen's test suites for LCG(2806196910506780709, 1, 2^{63})

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	92.74	37.86	28.38
	LEC02	40.72	8.17	56.93
Serial	LEC03	48.22	35.65	75.22
Gap	LEC04	81.04	3.92	12.54
	LEC05	24.46	77.22	36.98
	LEC06	30.86	45.53	51.56
Poker	LEC07	36.62	55.66	30.83
	LEC08	57.21	13.14	57.31
	LEC09	88.24	27.36	47.30
Coupon	LEC10	63.57	42.29	53.57
Permutation	LEC11	19.77	39.29	10.97
	LEC12	40.55	14.81	63.13
Runs-up	LEC13	33.41	52.91	61.23
Maximum of t	LEC14	74.45	29.21	80.80
Collision	LEC15	49.50	46.01	58.10
	LEC16	44.14	39.92	35.97
	LEC17	86.57	92.78	61.75
Vattulainen's test suite				
Equidistribution	VAT01	26.54	88.15	32.03
	VAT02	21.19	17.63	35.18
Serial	VAT03	45.69	41.45	24.86
Gap	VAT04	90.31	63.12	96.85
	VAT05	68.31	93.39	67.05
	VAT06	13.00	77.51	92.42
Runs-up	VAT07	33.41	52.91	61.23
Maximum of t	VAT08	**99.21**	14.08	98.85
	VAT09	57.81	**99.87**	81.39
Collision	VAT10	89.60	17.25	92.17
	VAT11	95.37	82.78	55.54
	VAT12	51.07	95.45	53.47

Table 19: Results of L'Ecuyer's and Vattulainen's test suites for LCG$(3249286849523012805, 1, 2^{63})$

Standard tests	Test ID	p-value (%)		
		Run 1	Run 2	Run 3
L'Ecuyer's test suite				
Equidistribution	LEC01	58.01	55.33	73.93
	LEC02	24.49	16.70	59.74
Serial	LEC03	15.71	66.87	16.38
Gap	LEC04	96.32	45.69	93.10
	LEC05	46.75	91.48	87.00
	LEC06	**99.75**	4.87	75.42
Poker	LEC07	35.74	36.17	27.53
	LEC08	43.87	2.44	81.31
	LEC09	95.64	22.13	58.70
Coupon	LEC10	29.65	39.55	40.70
Permutation	LEC11	20.87	88.72	66.01
	LEC12	56.71	94.88	55.82
Runs-up	LEC13	54.55	93.38	48.43
Maximum of t	LEC14	36.53	11.47	17.33
Collision	LEC15	2.75	40.92	63.38
	LEC16	76.24	71.87	96.60
	LEC17	56.34	87.44	**99.23**
Vattulainen's test suite				
Equidistribution	VAT01	15.14	47.45	**0.93**
	VAT02	69.21	25.57	36.92
Serial	VAT03	8.28	20.48	27.70
Gap	VAT04	63.31	94.24	88.31
	VAT05	31.39	22.10	49.37
	VAT06	13.01	46.26	43.77
Runs-up	VAT07	54.55	93.38	48.43
Maximum of t	VAT08	59.57	39.89	52.81
	VAT09	27.89	92.90	47.17
Collision	VAT10	14.14	94.70	98.35
	VAT11	44.42	7.91	73.09
	VAT12	65.50	6.63	16.15

We performed another standard tests with different parameters because the number of RNs tested with L'Ecuyer's and Vattulainen's ones is relatively small for 63-bit LCGs; $1.0 \times 10^7 \sim 2.0 \times 10^8$ for L'Ecuyer's, $6.0 \times 10^6 \sim 1.0 \times 10^9$ for Vattulainen's. The parameters are taken from Mascagni and Srinivasan's test suite [23]. Their tests were, however, performed for multiple RN sequences interleaved from different LCGs. Since we test a single RN sequence, we adjust the number of tested RNs so that it is about 1.0×10^{11}.

The standard tests with Mascagni and Srinivasan's parameters were performed basically only once for each LCGs because they require relatively long calculation time. Each test was repeated three times only when the first test was failed; the first p-value is less than 0.01 (1%) or larger than 0.99 (99%). Tables 20 $\sim$ 32 show the results of Mascagni and Srinivasan's the test suite. Some RNGs fail a test for the first subsequence but pass the test for the subsequent subsequences as shown in Table 33. Therefore, we consider that all the RNGs pass Mascagni and Srinivasan's test suite.

Table 20: Results of Mascagni and Srinivasan's test suite for LCG$(5^{19}, 0, 2^{48})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	1.84
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	85.19
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	76.46
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	47.55
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	12.01
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	19.60
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	94.70
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	54.21
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	2.25
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	**99.39**

N is the number of times the test was repeated for the (second-level) K-S test. n is the length of the random number sequence. Other parameters are described in Section 3.2.

Table 21: Results of Mascagni and Srinivasan's test suite for LCG$(5^{19}, 0, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	88.63
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	73.09
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	49.55
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	24.33
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	22.54
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	5.11
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	85.69
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	18.97
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	53.14
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	36.31

Table 22: Results of Mascagni and Srinivasan's test suite for LCG$(5^{23}, 0, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	30.53
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	81.58
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	11.85
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	83.83
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	49.36
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	32.60
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	9.19
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	13.32
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	94.20
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	87.14

Table 23: Results of Mascagni and Srinivasan's test suite for LCG($5^{25}, 0, 2^{63}$)

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	35.46
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	6.53
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	96.69
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	93.82
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	25.78
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	89.69
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	24.73
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	21.96
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	81.82
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	17.06

Table 24: Results of Mascagni and Srinivasan's test suite for LCG($5^{19}, 1, 2^{63}$)

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	1.70
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	47.08
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	42.43
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	19.55
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	95.33
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	8.31
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	74.36
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	83.08
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	51.17
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	42.04

Table 25: Results of Mascagni and Srinivasan's test suite for LCG$(5^{23}, 1, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	48.25
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	68.38
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	29.67
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	53.97
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	**0.18**
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	50.92
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	8.65
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	41.98
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	88.46
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	16.24

Table 26: Results of Mascagni and Srinivasan's test suite for LCG$(5^{25}, 1, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	93.43
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	**0.25**
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	11.45
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	92.79
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	15.04
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	53.21
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	77.31
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	55.16
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	84.32
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	57.70

Table 27: Results of Mascagni and Srinivasan's test suite for LCG($3512401965023503517, 0, 2^{63}$)

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	94.90
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	51.07
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	76.42
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	2.76
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	43.81
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	53.70
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	63.13
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	43.94
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	10.61
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	31.16

Table 28: Results of Mascagni and Srinivasan's test suite for LCG($2444805353187672469, 0, 2^{63}$)

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	60.11
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	51.87
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	9.05
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	98.24
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	4.14
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	42.91
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	24.05
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	21.23
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	36.45
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	97.41

Table 29: Results of Mascagni and Srinivasan's test suite for LCG(1987591058829310733, 0, 2^{63})

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	42.07
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	87.83
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	12.55
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	35.50
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	86.83
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	46.37
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	57.69
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	6.14
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	66.20
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	5.39

Table 30: Results of Mascagni and Srinivasan's test suite for LCG(9219741426499971445, 1, 2^{63})

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	85.38
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	74.15
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	65.03
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	94.35
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	31.26
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	53.11
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	17.55
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	62.03
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	11.37
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	10.55

Table 31: Results of Mascagni and Srinivasan's test suite for LCG$(2806196910506780709, 1, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	34.13
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	62.07
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	16.12
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	85.14
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	6.20
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	35.12
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	25.85
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	19.91
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	12.43
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	38.31

Table 32: Results of Mascagni and Srinivasan's test suite for LCG$(3249286849523012805, 1, 2^{63})$

Standard tests	Parameters	p-value
Equidistribution	$N = 5 \times 10^3, n = 2 \times 10^7, d = 10000$	42.55
Serial	$N = 10^3, n = 5 \times 10^7, d = 100$	51.10
Gap	$N = 10^3, n = 10^6, a = 0.50, b = 0.51, t = 200$	18.56
Poker	$N = 10^3, n = 10^7, k = 10, d = 10$	45.34
Coupon	$N = 10^3, n = 5 \times 10^6, d = 10, t = 39$	90.72
Permutation	$N = 10^3, n = 2 \times 10^7, t = 5$	96.23
Runs-up	$N = 10^3, n = 5 \times 10^7, t = 10$	69.42
Maximum of t	$N = 10^5, n = 5 \times 10^4, t = 16$	93.61
Collision 1	$N = 10^5, n = 10^5, \log md = 10, \log d = 3$	95.85
Collision 2	$N = 10^5, n = 2 \times 10^5, \log md = 4, \log d = 5$	84.81

Table 33: Results of additional tests for RNGs whose first subsequence failed

RNG	Failed test	p-value (%)		
		Run 1	Run 2	Run 3
LCG($5^{19}, 0, 2^{48}$)	Collision 2	**99.39**	52.80	6.83
LCG($5^{23}, 1, 2^{63}$)	Coupon	**0.18**	89.81	44.10
LCG($5^{25}, 1, 2^{63}$)	Serial	**0.25**	44.82	85.60

4.3 Results for DIEHARD test suite

The DIEHARD tests were also performed for all thirteen RNGs. For the tests, we set two significance levels depending on each test. In the case where a test returns more than five p-values, we set a significance level to 0.01 and consider that a RNG fails the test if we get six or more p-values less than 0.01 or more than 0.99. When a test returns more than two and less than six p-values, we consider that a RNG fails the test if all p-values are less than 0.01 or more than 0.99. When a test returns only one p-value, we set a significance level to 0.005. Namely, a RNG fails the test if the p-value is less than 0.005 or more than 0.995.

Tables 35 $\sim$ 47 shows the results of the DIEHARD tests. Since the name of each test is slightly long, it is designated for short as listed in Table 34. The p-values less than 0.01 or more than 0.99 are bold-faced.

The MCNP RNG (LCG($5^{19}, 0, 2^{48}$)) fails the OPSO, OQSO and DNA tests as shown in Table 35. In particular, less significant (lower) bits of RNs fail the tests. It is considered that these failures in less significant bits are caused by the shorter period than the significant bits as mentioned in Section 2.1. However, it does not seems that these failures have a significant impact in the practical use of the RNG.

On the other hand, all 63-bit LCGs pass all the tests though some p-values are less than 0.01 or more than 0.99. No failures are found in less significant bits for the OPSO, OQSO and DNA tests as found for the MCNP RNG.

Table 34: Short names for DIEHARD test suite

Full name	Short name
Birthday spacings test	BDAY
Overlapping 5-permutation test	OPERM
Binary rank test	RANK
Bitstream test	BSTREAM
Overlapping-pairs-sparse-occupancy test	OPSO
Overlapping-quadruples-sparse-occupancy test	OQSO
DNA test	DNA
Count-the-1's test on a stream of bytes	COUNT1S
Count-the-1's test for specific bytes	COUNT1B
Parking lot test	PARKING
Minimum distance test	MDIST
3-D sphere test	SPHERE
Squeeze test	SQUEEZE
Overlapping sums test	OSUMS
Runs test	RUNS
Craps test	CRAPS

Table 35: DIEHARD test results for LCG($5^{19}, 0, 2^{48}$)

Test		p-value
BDAY	bits 1 to 24	0.272790
	bits 2 to 25	0.821532
	bits 3 to 25	0.653590
	bits 4 to 25	0.147015
	bits 5 to 25	0.784672
	bits 6 to 25	0.340978
	bits 7 to 25	0.595325
	bits 8 to 25	0.014017
	bits 9 to 25	0.825536
K-S test for 9 p-values		0.115065
OPERM	1st	0.898243
	2nd	0.298070
RANK 31×31		0.347048
RANK 32×32		0.754761
RANK 6×8	bits 1 to 8	0.633029
	bits 2 to 9	0.527127
	bits 3 to 10	0.569367
	bits 4 to 11	0.569367
	bits 5 to 12	0.186756
	bits 6 to 13	0.208039
	bits 7 to 14	0.647876
	bits 8 to 15	0.849943
	bits 9 to 16	0.082948
	bits 10 to 17	0.102796
	bits 11 to 18	0.041357
	bits 12 to 19	0.770574
	bits 13 to 20	0.518207
	bits 14 to 21	**0.008043**
	bits 15 to 22	0.772758
	bits 16 to 23	0.230369
	bits 17 to 24	0.032800
	bits 18 to 25	0.821333
	bits 19 to 26	0.656534
	bits 20 to 27	0.545310
	bits 21 to 28	0.303901
	bits 22 to 29	0.129923
	bits 23 to 30	0.477979
	bits 24 to 31	0.031384
	bits 25 to 32	0.342400
K-S test for 25 p-values		0.891195
BSTREAM	1st	0.18773
	2nd	0.90955
	3rd	0.97771
	4th	0.31904
	5th	0.25549
	6th	0.20586
	7th	0.07795
	8th	0.37504
	9th	0.69037
	10th	0.38037
	11th	0.34964
	12th	0.62437
	13th	0.16768

Test		p-value
	14th	0.86830
	15th	0.71385
	16th	0.16885
	17th	0.36183
	18th	0.62082
	19th	0.14960
	20th	0.02271
OPSO	bits 23 to 32	**0.0000**
	bits 22 to 31	**0.0000**
	bits 21 to 30	**0.0000**
	bits 20 to 29	**0.0000**
	bits 19 to 28	**0.0001**
	bits 18 to 27	0.6639
	bits 17 to 26	0.0445
	bits 16 to 25	0.0125
	bits 15 to 24	0.7683
	bits 14 to 23	0.9712
	bits 13 to 22	0.1077
	bits 12 to 21	0.0717
	bits 11 to 20	0.7457
	bits 10 to 19	0.0598
	bits 9 to 18	0.1122
	bits 8 to 17	0.4597
	bits 7 to 16	**0.0011**
	bits 6 to 15	0.6319
	bits 5 to 14	0.7490
	bits 4 to 13	0.2914
	bits 3 to 12	0.1792
	bits 2 to 11	0.3253
	bits 1 to 10	0.7277
OQSO	bits 28 to 32	**1.0000**
	bits 27 to 31	**1.0000**
	bits 26 to 30	**1.0000**
	bits 25 to 29	**1.0000**
	bits 24 to 28	**1.0000**
	bits 23 to 27	**1.0000**
	bits 22 to 26	**0.0000**
	bits 21 to 25	**0.0000**
	bits 20 to 24	**0.0000**
	bits 19 to 23	0.1906
	bits 18 to 22	**0.0011**
	bits 17 to 21	0.3823
	bits 16 to 20	0.8394
	bits 15 to 19	0.2518
	bits 14 to 18	0.6487
	bits 13 to 17	0.5575
	bits 12 to 16	0.1634
	bits 11 to 15	0.6600
	bits 10 to 14	0.2096
	bits 9 to 13	0.3759
	bits 8 to 12	0.9191
	bits 7 to 11	0.8554
	bits 6 to 10	0.5535

Test		p-value
	bits 5 to 9	0.4955
	bits 4 to 8	0.0868
	bits 3 to 7	0.1943
	bits 2 to 6	0.8554
	bits 1 to 5	0.7421
DNA	bits 31 to 32	**1.0000**
	bits 30 to 31	**1.0000**
	bits 29 to 30	**1.0000**
	bits 28 to 29	**1.0000**
	bits 27 to 28	**1.0000**
	bits 26 to 27	0.1777
	bits 25 to 26	**0.0000**
	bits 24 to 25	**0.0000**
	bits 23 to 24	**0.0000**
	bits 22 to 23	**0.0000**
	bits 21 to 22	**0.0000**
	bits 20 to 21	0.4937
	bits 19 to 20	0.0613
	bits 18 to 19	0.2383
	bits 17 to 18	0.4831
	bits 16 to 17	0.0925
	bits 15 to 16	0.0197
	bits 14 to 15	0.7377
	bits 13 to 14	0.7171
	bits 12 to 13	0.0309
	bits 11 to 12	0.2803
	bits 10 to 11	0.8440
	bits 9 to 10	0.4550
	bits 8 to 9	0.4737
	bits 7 to 8	0.7834
	bits 6 to 7	0.4063
	bits 5 to 6	0.8959
	bits 4 to 5	0.3438
	bits 3 to 4	0.3972
	bits 2 to 3	0.8986
	bits 1 to 2	0.5407
COUNT1S	1st	0.681751
	2nd	0.255342
COUNT1B	bits 1 to 8	0.434733
	bits 2 to 9	0.718919
	bits 3 to 10	0.144793
	bits 4 to 11	0.685012
	bits 5 to 12	0.683909
	bits 6 to 13	0.502358
	bits 7 to 14	0.821357
	bits 8 to 15	0.375545
	bits 9 to 16	0.214134
	bits 10 to 17	0.735128
	bits 11 to 18	0.345899
	bits 12 to 19	0.798844
	bits 13 to 20	0.211146
	bits 14 to 21	0.301943
	bits 15 to 22	0.920976
	bits 16 to 23	0.579146
	bits 17 to 24	0.982771
	bits 18 to 25	0.316536
	bits 19 to 26	0.941200

Test		p-value
	bits 20 to 27	0.411558
	bits 21 to 28	0.542480
	bits 22 to 29	0.456693
	bits 23 to 30	0.308035
	bits 24 to 31	0.858280
	bits 25 to 32	0.759437
PARKING	1st	0.276387
	2nd	0.518210
	3rd	0.554479
	4th	0.590298
	5th	0.427537
	6th	0.146807
	7th	0.738676
	8th	0.554479
	9th	0.409702
	10th	0.954438
K-S test for 10 p-values		0.390666
MDIST		0.954438
SPHERE	1st	0.98097
	2nd	0.96610
	3rd	0.89832
	4th	0.54591
	5th	0.25548
	6th	0.23249
	7th	0.90286
	8th	0.68392
	9th	0.48022
	10th	0.83227
	11th	0.93155
	12th	0.29180
	13th	0.69449
	14th	0.45707
	15th	0.32792
	16th	0.23009
	17th	0.23249
	18th	0.30696
	19th	0.48874
	20th	0.39255
K-S test for 20 p-values		0.681575
SQUEEZE		0.439234
OSUMS	1st	0.579097
	2nd	0.426215
	3nd	0.748657
	4nd	0.347650
	5nd	0.349957
	6nd	0.625698
	7nd	0.381223
	8nd	0.496659
	9nd	0.754814
	10nd	0.121868
K-S test for 10 p-values		0.539464
RUNS	UP 1st	0.571959
	DOWN 1st	0.198622
	UP 2nd	0.776445
	DOWN 2nd	0.558044
CRAPS	No. of wins	0.909541
	Throws/game	0.049474

Table 36: DIEHARD test results for LCG$(5^{19}, 0, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.456631
	bits 2 to 25	0.934950
	bits 3 to 25	0.395226
	bits 4 to 25	0.151227
	bits 5 to 25	0.436915
	bits 6 to 25	0.881191
	bits 7 to 25	0.694738
	bits 8 to 25	0.630287
	bits 9 to 25	0.010339
K-S test for 9 p-values		0.052709
OPERM	1st	**0.997566**
	2nd	0.793837
RANK 31×31		0.588108
RANK 32×32		0.617617
RANK 6×8	bits 1 to 8	0.302278
	bits 2 to 9	0.904982
	bits 3 to 10	0.468827
	bits 4 to 11	0.540425
	bits 5 to 12	0.916199
	bits 6 to 13	0.816692
	bits 7 to 14	0.762551
	bits 8 to 15	0.225721
	bits 9 to 16	0.597547
	bits 10 to 17	0.116105
	bits 11 to 18	0.856230
	bits 12 to 19	0.951742
	bits 13 to 20	0.821750
	bits 14 to 21	0.042335
	bits 15 to 22	0.519765
	bits 16 to 23	0.465420
	bits 17 to 24	0.844583
	bits 18 to 25	0.815318
	bits 19 to 26	0.053148
	bits 20 to 27	0.914019
	bits 21 to 28	0.903223
	bits 22 to 29	0.475548
	bits 23 to 30	0.351186
	bits 24 to 31	0.100732
	bits 25 to 32	0.914019
K-S test for 25 p-values		0.681956
BSTREAM	1st	0.47082
	2nd	0.07200
	3rd	0.99618
	4th	0.86171
	5th	0.70343
	6th	0.97074
	7th	**0.00814**
	8th	0.64197
	9th	0.76317
	10th	0.70826
	11th	0.17420
	12th	0.01066
	13th	0.34792

Test		p-value
	14th	0.34189
	15th	0.32406
	16th	0.95865
	17th	0.18460
	18th	0.38572
	19th	0.50249
	20th	0.17905
OPSO	bits 23 to 32	0.7311
	bits 22 to 31	**0.0011**
	bits 21 to 30	0.6319
	bits 20 to 29	0.7490
	bits 19 to 28	0.2914
	bits 18 to 27	0.1792
	bits 17 to 26	0.3253
	bits 16 to 25	0.7277
	bits 15 to 24	0.5257
	bits 14 to 23	0.4913
	bits 13 to 22	0.8678
	bits 12 to 21	0.7673
	bits 11 to 20	0.5612
	bits 10 to 19	0.8377
	bits 9 to 18	0.4284
	bits 8 to 17	0.0658
	bits 7 to 16	0.2547
	bits 6 to 15	**0.9948**
	bits 5 to 14	0.9303
	bits 4 to 13	0.2670
	bits 3 to 12	0.6639
	bits 2 to 11	0.2843
	bits 1 to 10	0.3790
OQSO	bits 28 to 32	0.5575
	bits 27 to 31	0.1634
	bits 26 to 30	0.6600
	bits 25 to 29	0.2096
	bits 24 to 28	0.3759
	bits 23 to 27	0.9191
	bits 22 to 26	0.8554
	bits 21 to 25	0.5535
	bits 20 to 24	0.4955
	bits 19 to 23	0.0868
	bits 18 to 22	0.1943
	bits 17 to 21	0.8554
	bits 16 to 20	0.7421
	bits 15 to 19	0.9408
	bits 14 to 18	0.9062
	bits 13 to 17	0.2887
	bits 12 to 16	0.4190
	bits 11 to 15	0.3492
	bits 10 to 14	0.5588
	bits 9 to 13	0.9693
	bits 8 to 12	0.7377
	bits 7 to 11	0.6348
	bits 6 to 10	0.8912

Test		p-value
	bits 5 to 9	0.7829
	bits 4 to 8	0.9443
	bits 3 to 7	0.4456
	bits 2 to 6	0.8912
	bits 1 to 5	0.5225
DNA	bits 31 to 32	0.0925
	bits 30 to 31	0.0197
	bits 29 to 30	0.7377
	bits 28 to 29	0.7171
	bits 27 to 28	0.0309
	bits 26 to 27	0.2803
	bits 25 to 26	0.8440
	bits 24 to 25	0.4550
	bits 23 to 24	0.4737
	bits 22 to 23	0.7834
	bits 21 to 22	0.4063
	bits 20 to 21	0.8959
	bits 19 to 20	0.3438
	bits 18 to 19	0.3972
	bits 17 to 18	0.8986
	bits 16 to 17	0.5407
	bits 15 to 16	0.3624
	bits 14 to 15	0.9057
	bits 13 to 14	0.8468
	bits 12 to 13	0.7290
	bits 11 to 12	0.7019
	bits 10 to 11	0.8603
	bits 9 to 10	0.9227
	bits 8 to 9	0.6313
	bits 7 to 8	0.5020
	bits 6 to 7	0.8583
	bits 5 to 6	0.0732
	bits 4 to 5	0.2893
	bits 3 to 4	0.6833
	bits 2 to 3	0.2627
	bits 1 to 2	0.9101
COUNT1S	1st	0.360386
	2nd	**0.005499**
COUNT1B	bits 1 to 8	0.408927
	bits 2 to 9	0.737503
	bits 3 to 10	0.086679
	bits 4 to 11	0.885425
	bits 5 to 12	0.415990
	bits 6 to 13	0.414412
	bits 7 to 14	0.803623
	bits 8 to 15	0.080755
	bits 9 to 16	0.832648
	bits 10 to 17	0.916187
	bits 11 to 18	0.417992
	bits 12 to 19	0.888201
	bits 13 to 20	0.347871
	bits 14 to 21	0.744566
	bits 15 to 22	0.887958
	bits 16 to 23	0.467662
	bits 17 to 24	0.748463
	bits 18 to 25	0.088331
	bits 19 to 26	0.757319

Test		p-value
	bits 20 to 27	0.502216
	bits 21 to 28	0.702212
	bits 22 to 29	0.750895
	bits 23 to 30	0.445270
	bits 24 to 31	0.079477
	bits 25 to 32	0.755761
PARKING	1st	0.218799
	2nd	0.753306
	3rd	0.126820
	4th	0.050105
	5th	0.323972
	6th	0.708135
	7th	0.246694
	8th	0.276387
	9th	0.340551
	10th	0.659449
K-S test for 10 p-values		0.774103
MDIST		0.061572
SPHERE	1st	0.22119
	2nd	0.23493
	3rd	0.42664
	4th	0.68078
	5th	0.92590
	6th	0.63639
	7th	0.70631
	8th	0.88884
	9th	0.39730
	10th	0.56672
	11th	0.23983
	12th	**0.99167**
	13th	0.94178
	14th	0.39060
	15th	0.84633
	16th	0.57522
	17th	0.23271
	18th	0.40224
	19th	0.76420
	20th	0.28931
K-S test for 20 p-values		0.628216
SQUEEZE		0.181459
OSUMS	1st	0.483660
	2nd	0.782529
	3nd	0.561988
	4nd	0.310576
	5nd	0.273276
	6nd	0.194041
	7nd	0.111713
	8nd	0.095835
	9nd	0.622909
	10nd	0.215314
K-S test for 10 p-values		0.785766
RUNS	UP 1st	0.291501
	DOWN 1st	0.658321
	UP 2nd	0.819057
	DOWN 2nd	0.388523
CRAPS	No. of wins	0.516305
	Throws/game	0.622109

Table 37: DIEHARD test results for LCG($5^{23}, 0, 2^{63}$)

Test		p-value
BDAY	bits 1 to 24	0.510233
	bits 2 to 25	0.648150
	bits 3 to 25	0.647407
	bits 4 to 25	0.603159
	bits 5 to 25	0.931308
	bits 6 to 25	0.290724
	bits 7 to 25	0.039287
	bits 8 to 25	0.649477
	bits 9 to 25	0.966417
K-S test for 9 p-values		0.537279
OPERM	1st	**0.990270**
	2nd	0.812840
RANK 31×31		0.423292
RANK 32×32		0.343868
RANK 6×8	bits 1 to 8	0.396310
	bits 2 to 9	0.466788
	bits 3 to 10	0.940135
	bits 4 to 11	0.672198
	bits 5 to 12	0.250466
	bits 6 to 13	0.612980
	bits 7 to 14	0.444626
	bits 8 to 15	0.410735
	bits 9 to 16	0.564317
	bits 10 to 17	0.294100
	bits 11 to 18	0.450392
	bits 12 to 19	0.162234
	bits 13 to 20	0.055235
	bits 14 to 21	0.467041
	bits 15 to 22	0.041343
	bits 16 to 23	0.836969
	bits 17 to 24	0.659435
	bits 18 to 25	0.614567
	bits 19 to 26	0.738091
	bits 20 to 27	0.346112
	bits 21 to 28	0.877526
	bits 22 to 29	0.493602
	bits 23 to 30	0.304998
	bits 24 to 31	0.660920
	bits 25 to 32	0.105780
K-S test for 25 p-values		0.417022
BSTREAM	1st	0.81499
	2nd	0.15289
	3rd	0.10214
	4th	0.83252
	5th	0.76604
	6th	0.30745
	7th	0.28722
	8th	0.77455
	9th	0.58115
	10th	0.64197
	11th	0.55175
	12th	0.05757
	13th	**0.00977**

Test		p-value
	14th	0.41643
	15th	0.03959
	16th	0.91550
	17th	0.42099
	18th	0.04463
	19th	0.01368
	20th	0.63847
OPSO	bits 23 to 32	0.2216
	bits 22 to 31	0.4488
	bits 21 to 30	0.4844
	bits 20 to 29	0.7490
	bits 19 to 28	0.0289
	bits 18 to 27	0.8556
	bits 17 to 26	0.2592
	bits 16 to 25	0.7043
	bits 15 to 24	0.8444
	bits 14 to 23	0.7242
	bits 13 to 22	0.7577
	bits 12 to 21	0.6409
	bits 11 to 20	0.9216
	bits 10 to 19	0.8722
	bits 9 to 18	0.1431
	bits 8 to 17	0.9266
	bits 7 to 16	0.0114
	bits 6 to 15	0.2961
	bits 5 to 14	0.4406
	bits 4 to 13	0.5133
	bits 3 to 12	0.7019
	bits 2 to 11	0.6280
	bits 1 to 10	0.2237
OQSO	bits 28 to 32	0.7779
	bits 27 to 31	0.4523
	bits 26 to 30	0.5454
	bits 25 to 29	0.0290
	bits 24 to 28	0.4177
	bits 23 to 27	0.9863
	bits 22 to 26	0.0283
	bits 21 to 25	0.3355
	bits 20 to 24	0.5986
	bits 19 to 23	0.0176
	bits 18 to 22	0.0323
	bits 17 to 21	0.2853
	bits 16 to 20	0.8403
	bits 15 to 19	0.5615
	bits 14 to 18	0.1833
	bits 13 to 17	0.7421
	bits 12 to 16	0.3442
	bits 11 to 15	0.9416
	bits 10 to 14	0.9443
	bits 9 to 13	0.8451
	bits 8 to 12	0.5171
	bits 7 to 11	0.1888
	bits 6 to 10	0.1155

Test		p-value
	bits 5 to 9	0.0709
	bits 4 to 8	**0.0024**
	bits 3 to 7	0.9264
	bits 2 to 6	0.9021
	bits 1 to 5	0.6475
DNA	bits 31 to 32	0.1159
	bits 30 to 31	0.3025
	bits 29 to 30	0.6457
	bits 28 to 29	0.6468
	bits 27 to 28	0.8888
	bits 26 to 27	0.6457
	bits 25 to 26	0.4843
	bits 24 to 25	0.4773
	bits 23 to 24	0.6999
	bits 22 to 23	0.5008
	bits 21 to 22	0.2984
	bits 20 to 21	0.3223
	bits 19 to 20	0.6927
	bits 18 to 19	0.7191
	bits 17 to 18	0.0882
	bits 16 to 17	0.9698
	bits 15 to 16	0.8609
	bits 14 to 15	0.1649
	bits 13 to 14	0.8954
	bits 12 to 13	0.0965
	bits 11 to 12	0.8397
	bits 10 to 11	**0.9996**
	bits 9 to 10	0.6566
	bits 8 to 9	0.2100
	bits 7 to 8	0.4363
	bits 6 to 7	0.4914
	bits 5 to 6	0.6077
	bits 4 to 5	**0.9996**
	bits 3 to 4	0.3066
	bits 2 to 3	0.7538
	bits 1 to 2	0.8331
COUNT1S	1st	0.491810
	2nd	0.147809
COUNT1B	bits 1 to 8	0.547158
	bits 2 to 9	0.878889
	bits 3 to 10	0.050935
	bits 4 to 11	0.511621
	bits 5 to 12	0.316532
	bits 6 to 13	0.492223
	bits 7 to 14	0.789981
	bits 8 to 15	0.383586
	bits 9 to 16	0.607016
	bits 10 to 17	0.146271
	bits 11 to 18	0.957428
	bits 12 to 19	0.654990
	bits 13 to 20	0.870018
	bits 14 to 21	0.334043
	bits 15 to 22	0.620158
	bits 16 to 23	0.483766
	bits 17 to 24	0.973638
	bits 18 to 25	0.192997
	bits 19 to 26	0.391369

Test		p-value
	bits 20 to 27	0.671128
	bits 21 to 28	0.942104
	bits 22 to 29	0.462529
	bits 23 to 30	0.177982
	bits 24 to 31	0.362327
	bits 25 to 32	0.853652
PARKING	1st	0.794438
	2nd	0.015932
	3rd	0.033889
	4th	0.590298
	5th	0.374623
	6th	0.392053
	7th	0.659449
	8th	**0.997991**
	9th	0.969407
	10th	0.987371
K-S test for 10 p-values		0.914198
MDIST		0.518994
SPHERE	1st	0.39297
	2nd	0.97443
	3rd	0.91314
	4th	0.24055
	5th	0.57765
	6th	0.71839
	7th	0.86598
	8th	0.03872
	9th	0.50686
	10th	0.57368
	11th	0.50174
	12th	0.90182
	13th	0.50860
	14th	0.58902
	15th	0.84334
	16th	0.48498
	17th	0.04574
	18th	0.47912
	19th	0.98710
	20th	0.65813
K-S test for 20 p-values		0.803715
SQUEEZE		0.339558
OSUMS	1st	0.783339
	2nd	0.981177
	3nd	0.929536
	4nd	0.253271
	5nd	0.420325
	6nd	0.065410
	7nd	0.858929
	8nd	0.514356
	9nd	0.370627
	10nd	0.248515
K-S test for 10 p-values		0.187704
RUNS	UP 1st	0.297819
	DOWN 1st	0.107353
	UP 2nd	0.603812
	DOWN 2nd	0.683517
CRAPS	No. of wins	0.892426
	Throws/game	0.892426

Table 38: DIEHARD test results for LCG($5^{25}, 0, 2^{63}$)

Test		p-value
BDAY	bits 1 to 24	0.749158
	bits 2 to 25	0.289727
	bits 3 to 25	**0.006385**
	bits 4 to 25	0.225392
	bits 5 to 25	0.673204
	bits 6 to 25	0.513489
	bits 7 to 25	0.881011
	bits 8 to 25	0.949979
	bits 9 to 25	0.872984
K-S test for 9 p-values		0.452078
OPERM	1st	0.021604
	2nd	0.303615
RANK 31×31		0.761668
RANK 32×32		0.432260
RANK 6×8	bits 1 to 8	0.095827
	bits 2 to 9	0.218704
	bits 3 to 10	0.108018
	bits 4 to 11	0.544305
	bits 5 to 12	**0.004031**
	bits 6 to 13	0.831863
	bits 7 to 14	0.141851
	bits 8 to 15	0.821328
	bits 9 to 16	0.534020
	bits 10 to 17	0.374278
	bits 11 to 18	0.325640
	bits 12 to 19	**0.993534**
	bits 13 to 20	0.958890
	bits 14 to 21	0.818070
	bits 15 to 22	0.984705
	bits 16 to 23	0.416342
	bits 17 to 24	0.166343
	bits 18 to 25	0.320967
	bits 19 to 26	0.482612
	bits 20 to 27	0.039089
	bits 21 to 28	0.106775
	bits 22 to 29	0.834839
	bits 23 to 30	0.555757
	bits 24 to 31	0.035217
	bits 25 to 32	0.073664
K-S test for 25 p-values		0.866133
BSTREAM	1st	0.54712
	2nd	0.83311
	3rd	0.78081
	4th	0.77665
	5th	0.65152
	6th	0.61190
	7th	**0.00730**
	8th	0.43382
	9th	0.58844
	10th	0.17783
	11th	0.93723
	12th	0.60022
	13th	0.78968

Test		p-value
	14th	0.88405
	15th	0.61904
	16th	0.94746
	17th	0.08536
	18th	0.25700
	19th	0.16246
	20th	**0.00416**
OPSO	bits 23 to 32	0.3241
	bits 22 to 31	0.4433
	bits 21 to 30	0.1064
	bits 20 to 29	0.8166
	bits 19 to 28	0.9757
	bits 18 to 27	0.6788
	bits 17 to 26	0.8229
	bits 16 to 25	0.7609
	bits 15 to 24	0.6069
	bits 14 to 23	0.1162
	bits 13 to 22	0.6227
	bits 12 to 21	0.7208
	bits 11 to 20	0.8533
	bits 10 to 19	0.4176
	bits 9 to 18	0.9308
	bits 8 to 17	0.7043
	bits 7 to 16	0.5064
	bits 6 to 15	0.3518
	bits 5 to 14	0.0335
	bits 4 to 13	0.3909
	bits 3 to 12	0.8736
	bits 2 to 11	0.6601
	bits 1 to 10	0.9550
OQSO	bits 28 to 32	0.9621
	bits 27 to 31	0.1168
	bits 26 to 30	0.0664
	bits 25 to 29	0.4928
	bits 24 to 28	0.5722
	bits 23 to 27	0.6796
	bits 22 to 26	0.1358
	bits 21 to 25	0.5090
	bits 20 to 24	0.5735
	bits 19 to 23	0.8215
	bits 18 to 22	0.7888
	bits 17 to 21	0.5252
	bits 16 to 20	0.9737
	bits 15 to 19	0.7718
	bits 14 to 18	0.2683
	bits 13 to 17	0.2475
	bits 12 to 16	0.9640
	bits 11 to 15	0.7697
	bits 10 to 14	0.7550
	bits 9 to 13	0.3293
	bits 8 to 12	0.0482
	bits 7 to 11	0.0437
	bits 6 to 10	0.4032

Test		p-value
	bits 5 to 9	0.7769
	bits 4 to 8	0.4177
	bits 3 to 7	0.1418
	bits 2 to 6	0.9462
	bits 1 to 5	0.7058
DNA	bits 31 to 32	0.8218
	bits 30 to 31	0.8404
	bits 29 to 30	0.6324
	bits 28 to 29	0.9716
	bits 27 to 28	0.0334
	bits 26 to 27	0.2793
	bits 25 to 26	0.0483
	bits 24 to 25	0.6927
	bits 23 to 24	0.3769
	bits 22 to 23	0.9443
	bits 21 to 22	0.5442
	bits 20 to 21	0.4902
	bits 19 to 20	0.1792
	bits 18 to 19	0.6958
	bits 17 to 18	0.2569
	bits 16 to 17	0.8440
	bits 15 to 16	0.8730
	bits 14 to 15	0.5477
	bits 13 to 14	0.3984
	bits 12 to 13	0.5803
	bits 11 to 12	0.7720
	bits 10 to 11	0.1627
	bits 9 to 10	0.4410
	bits 8 to 9	0.4086
	bits 7 to 8	0.6822
	bits 6 to 7	0.6770
	bits 5 to 6	0.0548
	bits 4 to 5	0.3927
	bits 3 to 4	0.4831
	bits 2 to 3	0.1032
	bits 1 to 2	0.0305
COUNT1S	1st	0.238104
	2nd	0.654703
COUNT1B	bits 1 to 8	0.332852
	bits 2 to 9	0.904618
	bits 3 to 10	0.622997
	bits 4 to 11	0.873031
	bits 5 to 12	**0.998515**
	bits 6 to 13	0.051583
	bits 7 to 14	0.385513
	bits 8 to 15	0.154935
	bits 9 to 16	0.965408
	bits 10 to 17	0.100266
	bits 11 to 18	0.465014
	bits 12 to 19	0.931173
	bits 13 to 20	0.871369
	bits 14 to 21	0.315702
	bits 15 to 22	0.746001
	bits 16 to 23	0.373761
	bits 17 to 24	0.550210
	bits 18 to 25	0.062564
	bits 19 to 26	0.320470

Test		p-value
	bits 20 to 27	0.925802
	bits 21 to 28	0.474846
	bits 22 to 29	0.175687
	bits 23 to 30	0.751236
	bits 24 to 31	0.860441
	bits 25 to 32	0.970178
PARKING	1st	0.323972
	2nd	0.708135
	3rd	0.323972
	4th	0.958644
	5th	0.659449
	6th	0.853193
	7th	0.899470
	8th	0.781201
	9th	0.969407
	10th	0.977738
K-S test for 10 p-values		**0.993381**
MDIST		0.407511
SPHERE	1st	0.54685
	2nd	0.62404
	3rd	0.28389
	4th	0.93359
	5th	0.35013
	6th	0.25624
	7th	0.61072
	8th	**0.00332**
	9th	0.50748
	10th	0.99591
	11th	0.67114
	12th	0.10592
	13th	0.18394
	14th	0.74981
	15th	0.68083
	16th	0.69253
	17th	0.29552
	18th	0.65892
	19th	0.05535
	20th	0.58649
K-S test for 20 p-values		0.243988
SQUEEZE		0.896761
OSUMS	1st	0.937360
	2nd	0.748848
	3nd	0.817578
	4nd	0.506994
	5nd	0.558444
	6nd	0.397806
	7nd	0.341894
	8nd	0.765528
	9nd	0.691076
	10nd	0.225903
K-S test for 10 p-values		0.582238
RUNS	UP 1st	0.488985
	DOWN 1st	0.780775
	UP 2nd	0.733830
	DOWN 2nd	0.489666
CRAPS	No. of wins	0.974980
	Throws/game	0.772641

Table 39: DIEHARD test results for LCG($5^{19}, 1, 2^{63}$)

Test		p-value
BDAY	bits 1 to 24	0.598890
	bits 2 to 25	0.217354
	bits 3 to 25	0.812984
	bits 4 to 25	0.221838
	bits 5 to 25	0.211322
	bits 6 to 25	0.273215
	bits 7 to 25	0.966747
	bits 8 to 25	0.501631
	bits 9 to 25	0.214666
K-S test for 9 p-values		0.527559
OPERM	1st	0.185146
	2nd	0.397079
RANK 31×31		0.422641
RANK 32×32		0.757807
RANK 6×8	bits 1 to 8	0.526444
	bits 2 to 9	0.283043
	bits 3 to 10	0.885177
	bits 4 to 11	0.470605
	bits 5 to 12	0.158696
	bits 6 to 13	0.813975
	bits 7 to 14	0.223295
	bits 8 to 15	0.156549
	bits 9 to 16	0.191216
	bits 10 to 17	0.882014
	bits 11 to 18	0.690904
	bits 12 to 19	0.967506
	bits 13 to 20	0.947775
	bits 14 to 21	0.780829
	bits 15 to 22	0.744143
	bits 16 to 23	0.024331
	bits 17 to 24	0.060709
	bits 18 to 25	0.739415
	bits 19 to 26	0.163587
	bits 20 to 27	0.518397
	bits 21 to 28	0.786544
	bits 22 to 29	0.615802
	bits 23 to 30	0.596588
	bits 24 to 31	**0.996263**
	bits 25 to 32	0.308065
K-S test for 25 p-values		0.334937
BSTREAM	1st	0.94517
	2nd	0.17541
	3rd	0.79503
	4th	0.22499
	5th	0.74701
	6th	0.62526
	7th	0.55082
	8th	0.25850
	9th	0.43382
	10th	0.09524
	11th	0.39379
	12th	0.30335
	13th	0.52578
	14th	0.61190
	15th	0.47268
	16th	0.05262
	17th	0.71543
	18th	0.38126
	19th	0.84675
	20th	0.92452
OPSO	bits 23 to 32	0.9423
	bits 22 to 31	0.7055
	bits 21 to 30	0.7219
	bits 20 to 29	0.8110
	bits 19 to 28	0.0269
	bits 18 to 27	0.0454
	bits 17 to 26	0.5571
	bits 16 to 25	0.0298
	bits 15 to 24	0.5639
	bits 14 to 23	0.4447
	bits 13 to 22	0.5353
	bits 12 to 21	0.3949
	bits 11 to 20	0.0510
	bits 10 to 19	0.6763
	bits 9 to 18	0.0413
	bits 8 to 17	0.7694
	bits 7 to 16	0.9266
	bits 6 to 15	0.2750
	bits 5 to 14	0.2750
	bits 4 to 13	0.3582
	bits 3 to 12	0.8893
	bits 2 to 11	0.4570
	bits 1 to 10	0.3404
OQSO	bits 28 to 32	0.8369
	bits 27 to 31	0.4216
	bits 26 to 30	0.3417
	bits 25 to 29	0.0270
	bits 24 to 28	0.3220
	bits 23 to 27	0.1659
	bits 22 to 26	0.8079
	bits 21 to 25	0.6450
	bits 20 to 24	0.8088
	bits 19 to 23	0.5986
	bits 18 to 22	0.0137
	bits 17 to 21	0.7497
	bits 16 to 20	0.9045
	bits 15 to 19	0.9733
	bits 14 to 18	0.1488
	bits 13 to 17	**0.9922**
	bits 12 to 16	0.3220
	bits 11 to 15	0.1851
	bits 10 to 14	0.0963
	bits 9 to 13	0.7208
	bits 8 to 12	0.3914
	bits 7 to 11	0.7208
	bits 6 to 10	0.5279

Test		p-value
	bits 5 to 9	0.1209
	bits 4 to 8	**0.0049**
	bits 3 to 7	0.1511
	bits 2 to 6	0.3028
	bits 1 to 5	0.7000
DNA	bits 31 to 32	**0.9933**
	bits 30 to 31	0.9463
	bits 29 to 30	**0.0021**
	bits 28 to 29	0.0421
	bits 27 to 28	0.4271
	bits 26 to 27	0.4749
	bits 25 to 26	0.4456
	bits 24 to 25	0.0558
	bits 23 to 24	0.8766
	bits 22 to 23	0.4808
	bits 21 to 22	0.5302
	bits 20 to 21	0.3558
	bits 19 to 20	0.0149
	bits 18 to 19	0.8736
	bits 17 to 18	0.3893
	bits 16 to 17	0.2627
	bits 15 to 16	0.0181
	bits 14 to 15	0.5814
	bits 13 to 14	0.6566
	bits 12 to 13	0.2852
	bits 11 to 12	0.1500
	bits 10 to 11	0.5998
	bits 9 to 10	0.7538
	bits 8 to 9	0.0863
	bits 7 to 8	0.6413
	bits 6 to 7	0.8674
	bits 5 to 6	0.8390
	bits 4 to 5	0.5699
	bits 3 to 4	0.0201
	bits 2 to 3	0.9419
	bits 1 to 2	0.5442
COUNT1S	1st	0.905273
	2nd	0.045513
COUNT1B	bits 1 to 8	0.987626
	bits 2 to 9	0.027794
	bits 3 to 10	0.347403
	bits 4 to 11	0.081196
	bits 5 to 12	0.209411
	bits 6 to 13	0.152336
	bits 7 to 14	0.224502
	bits 8 to 15	0.602737
	bits 9 to 16	0.320630
	bits 10 to 17	0.564691
	bits 11 to 18	0.967510
	bits 12 to 19	0.110182
	bits 13 to 20	0.218207
	bits 14 to 21	0.605838
	bits 15 to 22	0.621087
	bits 16 to 23	0.502065
	bits 17 to 24	0.692466
	bits 18 to 25	0.194252
	bits 19 to 26	0.924246

Test		p-value
	bits 20 to 27	0.360840
	bits 21 to 28	0.532679
	bits 22 to 29	0.830430
	bits 23 to 30	0.308146
	bits 24 to 31	0.269508
	bits 25 to 32	0.629817
PARKING	1st	0.078457
	2nd	0.192812
	3rd	0.807188
	4th	0.126820
	5th	0.117571
	6th	0.009936
	7th	0.518210
	8th	0.873180
	9th	0.914635
	10th	0.842447
K-S test for 10 p-values		0.746309
MDIST		0.214052
SPHERE	1st	0.54033
	2nd	0.90737
	3rd	0.01103
	4th	0.51826
	5th	0.14380
	6th	0.60742
	7th	0.43446
	8th	0.42220
	9th	0.10220
	10th	0.05250
	11th	0.27437
	12th	0.80002
	13th	0.42436
	14th	0.03394
	15th	0.20920
	16th	0.89106
	17th	0.56395
	18th	0.82387
	19th	0.86696
	20th	0.67515
K-S test for 20 p-values		0.218178
SQUEEZE		0.073604
OSUMS	1st	0.757899
	2nd	0.717099
	3nd	0.263340
	4nd	0.403221
	5nd	0.149960
	6nd	0.965015
	7nd	0.971890
	8nd	0.398787
	9nd	0.516020
	10nd	0.889466
K-S test for 10 p-values		0.616520
RUNS	UP 1st	0.441220
	DOWN 1st	0.910215
	UP 2nd	0.715504
	DOWN 2nd	0.091807
CRAPS	No. of wins	0.274687
	Throws/game	0.460578

Table 40: DIEHARD test results for LCG$(5^{23}, 1, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	**0.990853**
	bits 2 to 25	0.661486
	bits 3 to 25	0.417590
	bits 4 to 25	0.453951
	bits 5 to 25	0.712944
	bits 6 to 25	0.913126
	bits 7 to 25	0.700820
	bits 8 to 25	0.530341
	bits 9 to 25	0.866129
K-S test for 9 p-values		0.933205
OPERM	1st	0.497157
	2nd	0.122773
RANK 31×31		0.330828
RANK 32×32		0.556899
RANK 6×8	bits 1 to 8	0.058063
	bits 2 to 9	0.663510
	bits 3 to 10	0.950939
	bits 4 to 11	0.288193
	bits 5 to 12	0.078874
	bits 6 to 13	0.723909
	bits 7 to 14	0.982357
	bits 8 to 15	0.137520
	bits 9 to 16	0.161401
	bits 10 to 17	0.197574
	bits 11 to 18	0.967418
	bits 12 to 19	0.836184
	bits 13 to 20	0.552174
	bits 14 to 21	0.590520
	bits 15 to 22	0.416323
	bits 16 to 23	0.259904
	bits 17 to 24	0.910969
	bits 18 to 25	0.434705
	bits 19 to 26	0.972342
	bits 20 to 27	0.376817
	bits 21 to 28	0.286520
	bits 22 to 29	0.972737
	bits 23 to 30	0.148462
	bits 24 to 31	0.708309
	bits 25 to 32	0.925197
K-S test for 25 p-values		0.789528
BSTREAM	1st	0.10090
	2nd	0.25026
	3rd	0.75073
	4th	0.39649
	5th	0.94199
	6th	0.22081
	7th	0.17844
	8th	0.19862
	9th	0.05040
	10th	**0.99501**
	11th	0.52298
	12th	0.55636
	13th	0.82600
	14th	0.86325
	15th	0.81499
	16th	0.82899
	17th	0.38572
	18th	0.91367
	19th	0.97927
	20th	0.77804
OPSO	bits 23 to 32	0.0586
	bits 22 to 31	0.0485
	bits 21 to 30	0.4122
	bits 20 to 29	0.9606
	bits 19 to 28	0.1677
	bits 18 to 27	0.2085
	bits 17 to 26	0.2592
	bits 16 to 25	0.3493
	bits 15 to 24	0.2536
	bits 14 to 23	0.0859
	bits 13 to 22	0.1175
	bits 12 to 21	0.2558
	bits 11 to 20	0.8184
	bits 10 to 19	0.9383
	bits 9 to 18	0.3442
	bits 8 to 17	0.6874
	bits 7 to 16	0.7067
	bits 6 to 15	**0.9978**
	bits 5 to 14	0.3896
	bits 4 to 13	0.7746
	bits 3 to 12	0.8386
	bits 2 to 11	0.1245
	bits 1 to 10	0.2405
OQSO	bits 28 to 32	0.7242
	bits 27 to 31	0.7000
	bits 26 to 30	0.1209
	bits 25 to 29	0.1701
	bits 24 to 28	0.9462
	bits 23 to 27	0.7572
	bits 22 to 26	0.9062
	bits 21 to 25	0.9525
	bits 20 to 24	0.5642
	bits 19 to 23	0.2265
	bits 18 to 22	0.5090
	bits 17 to 21	0.8411
	bits 16 to 20	0.5854
	bits 15 to 19	0.0514
	bits 14 to 18	0.0412
	bits 13 to 17	0.9683
	bits 12 to 16	0.7174
	bits 11 to 15	0.6796
	bits 10 to 14	0.1472
	bits 9 to 13	0.5212
	bits 8 to 12	0.6168
	bits 7 to 11	0.7366
	bits 6 to 10	0.2225

Test		p-value
	bits 5 to 9	0.1719
	bits 4 to 8	0.7058
	bits 3 to 7	0.2981
	bits 2 to 6	0.4097
	bits 1 to 5	0.6051
DNA	bits 31 to 32	0.9206
	bits 30 to 31	0.1761
	bits 29 to 30	0.3769
	bits 28 to 29	0.3525
	bits 27 to 28	0.1521
	bits 26 to 27	0.4410
	bits 25 to 26	0.6357
	bits 24 to 25	0.3384
	bits 23 to 24	0.6457
	bits 22 to 23	0.9309
	bits 21 to 22	0.1335
	bits 20 to 21	0.3927
	bits 19 to 20	0.8871
	bits 18 to 19	0.2292
	bits 17 to 18	0.1528
	bits 16 to 17	0.4433
	bits 15 to 16	0.9630
	bits 14 to 15	0.1108
	bits 13 to 14	0.5384
	bits 12 to 13	0.1148
	bits 11 to 12	0.9618
	bits 10 to 11	0.9888
	bits 9 to 10	0.8536
	bits 8 to 9	0.3893
	bits 7 to 8	0.3118
	bits 6 to 7	0.3171
	bits 5 to 6	0.1694
	bits 4 to 5	0.6446
	bits 3 to 4	0.0268
	bits 2 to 3	0.5008
	bits 1 to 2	0.4843
COUNT1S	1st	0.679839
	2nd	0.680702
COUNT1B	bits 1 to 8	0.984081
	bits 2 to 9	0.617778
	bits 3 to 10	0.296050
	bits 4 to 11	0.272029
	bits 5 to 12	0.886256
	bits 6 to 13	0.740017
	bits 7 to 14	0.790192
	bits 8 to 15	0.425556
	bits 9 to 16	0.436191
	bits 10 to 17	0.085636
	bits 11 to 18	0.588265
	bits 12 to 19	**0.992207**
	bits 13 to 20	0.808170
	bits 14 to 21	0.763816
	bits 15 to 22	0.106782
	bits 16 to 23	0.057482
	bits 17 to 24	0.841543
	bits 18 to 25	0.647273
	bits 19 to 26	0.317607

Test		p-value
	bits 20 to 27	0.453169
	bits 21 to 28	0.073325
	bits 22 to 29	0.176200
	bits 23 to 30	0.361816
	bits 24 to 31	0.440753
	bits 25 to 32	0.367757
PARKING	1st	0.374623
	2nd	0.246694
	3rd	0.078457
	4th	0.445521
	5th	0.374623
	6th	0.232514
	7th	0.692266
	8th	0.590298
	9th	0.126820
	10th	0.340551
K-S test for 10 p-values		0.874748
MDIST		0.756212
SPHERE	1st	**0.00725**
	2nd	0.93088
	3rd	0.43462
	4th	0.50485
	5th	0.80240
	6th	0.35580
	7th	0.14443
	8th	0.14023
	9th	0.30040
	10th	0.59817
	11th	0.92163
	12th	0.02845
	13th	0.30271
	14th	0.30549
	15th	0.62047
	16th	0.94403
	17th	0.80386
	18th	0.79112
	19th	0.42274
	20th	0.38949
K-S test for 20 p-values		0.091389
SQUEEZE		0.968048
OSUMS	1st	0.282576
	2nd	0.675938
	3nd	0.872226
	4nd	0.739651
	5nd	0.385764
	6nd	0.168696
	7nd	0.303410
	8nd	0.572098
	9nd	0.558191
	10nd	0.460775
K-S test for 10 p-values		0.271523
RUNS	UP 1st	0.084616
	DOWN 1st	0.264600
	UP 2nd	0.099764
	DOWN 2nd	0.078459
CRAPS	No. of wins	0.761696
	Throws/game	0.564042

Table 41: DIEHARD test results for LCG($5^{25}, 1, 2^{63}$)

Test		p-value
BDAY	bits 1 to 24	0.570413
	bits 2 to 25	0.031149
	bits 3 to 25	0.480883
	bits 4 to 25	0.472966
	bits 5 to 25	0.044630
	bits 6 to 25	0.983694
	bits 7 to 25	0.029639
	bits 8 to 25	0.347548
	bits 9 to 25	0.826033
K-S test for 9 p-values		0.753745
OPERM	1st	0.960472
	2nd	**0.990484**
RANK 31×31		0.912279
RANK 32×32		0.866558
RANK 6×8	bits 1 to 8	0.311968
	bits 2 to 9	0.848220
	bits 3 to 10	0.818828
	bits 4 to 11	0.982629
	bits 5 to 12	0.927205
	bits 6 to 13	0.664633
	bits 7 to 14	0.152632
	bits 8 to 15	0.236126
	bits 9 to 16	0.862460
	bits 10 to 17	0.441143
	bits 11 to 18	0.939231
	bits 12 to 19	0.851465
	bits 13 to 20	0.798104
	bits 14 to 21	0.978214
	bits 15 to 22	0.559749
	bits 16 to 23	0.185878
	bits 17 to 24	0.157266
	bits 18 to 25	0.473947
	bits 19 to 26	**0.002875**
	bits 20 to 27	0.129923
	bits 21 to 28	**0.003189**
	bits 22 to 29	0.745328
	bits 23 to 30	0.716229
	bits 24 to 31	0.207200
	bits 25 to 32	0.234449
K-S test for 25 p-values		0.708471
BSTREAM	1st	0.48851
	2nd	0.82359
	3rd	0.73720
	4th	0.79102
	5th	0.41187
	6th	0.98868
	7th	0.70262
	8th	**0.00252**
	9th	0.28882
	10th	0.48385
	11th	0.78356
	12th	0.76890
	13th	0.02322

Test		p-value
	14th	0.41825
	15th	0.43750
	16th	0.25549
	17th	0.93752
	18th	0.48106
	19th	0.94944
	20th	0.09643
OPSO	bits 23 to 32	0.6639
	bits 22 to 31	0.4570
	bits 21 to 30	0.5775
	bits 20 to 29	0.3557
	bits 19 to 28	0.2648
	bits 18 to 27	0.4393
	bits 17 to 26	0.0250
	bits 16 to 25	0.2902
	bits 15 to 24	0.8343
	bits 14 to 23	0.8840
	bits 13 to 22	0.9251
	bits 12 to 21	0.4190
	bits 11 to 20	0.6701
	bits 10 to 19	0.8452
	bits 9 to 18	0.8887
	bits 8 to 17	0.9506
	bits 7 to 16	0.6525
	bits 6 to 15	0.1559
	bits 5 to 14	0.1245
	bits 4 to 13	0.4502
	bits 3 to 12	0.2076
	bits 2 to 11	0.1347
	bits 1 to 10	0.2648
OQSO	bits 28 to 32	0.5722
	bits 27 to 31	0.0112
	bits 26 to 30	0.7421
	bits 25 to 29	0.0728
	bits 24 to 28	0.4497
	bits 23 to 27	0.7748
	bits 22 to 26	0.4137
	bits 21 to 25	0.3355
	bits 20 to 24	0.1511
	bits 19 to 23	0.1142
	bits 18 to 22	0.4631
	bits 17 to 21	**0.9955**
	bits 16 to 20	0.9466
	bits 15 to 19	0.2922
	bits 14 to 18	0.6674
	bits 13 to 17	0.7343
	bits 12 to 16	0.7388
	bits 11 to 15	0.4163
	bits 10 to 14	0.2106
	bits 9 to 13	0.3159
	bits 8 to 12	0.5252
	bits 7 to 11	0.1906
	bits 6 to 10	0.5077

Test		p-value
	bits 5 to 9	0.4833
	bits 4 to 8	0.2887
	bits 3 to 7	0.0554
	bits 2 to 6	0.6259
	bits 1 to 5	0.1202
DNA	bits 31 to 32	0.8316
	bits 30 to 31	0.8883
	bits 29 to 30	0.2355
	bits 28 to 29	0.4503
	bits 27 to 28	0.3961
	bits 26 to 27	0.8642
	bits 25 to 26	0.5536
	bits 24 to 25	0.7309
	bits 23 to 24	0.3514
	bits 22 to 23	0.1001
	bits 21 to 22	0.7453
	bits 20 to 21	0.6641
	bits 19 to 20	0.4224
	bits 18 to 19	0.0753
	bits 17 to 18	0.5384
	bits 16 to 17	0.8616
	bits 15 to 16	0.9106
	bits 14 to 15	0.9698
	bits 13 to 14	0.2008
	bits 12 to 13	0.3836
	bits 11 to 12	0.6134
	bits 10 to 11	0.4855
	bits 9 to 10	0.7928
	bits 8 to 9	0.6706
	bits 7 to 8	0.9052
	bits 6 to 7	0.2903
	bits 5 to 6	0.5255
	bits 4 to 5	0.0665
	bits 3 to 4	0.1649
	bits 2 to 3	0.2714
	bits 1 to 2	0.0086
COUNT1S	1st	0.105943
	2nd	0.921093
COUNT1B	bits 1 to 8	0.635437
	bits 2 to 9	0.595580
	bits 3 to 10	0.103928
	bits 4 to 11	0.259409
	bits 5 to 12	0.399723
	bits 6 to 13	0.599593
	bits 7 to 14	0.529607
	bits 8 to 15	0.514767
	bits 9 to 16	0.462789
	bits 10 to 17	0.400949
	bits 11 to 18	0.373807
	bits 12 to 19	0.213108
	bits 13 to 20	0.853634
	bits 14 to 21	0.569019
	bits 15 to 22	0.068772
	bits 16 to 23	0.203534
	bits 17 to 24	0.263829
	bits 18 to 25	0.445256
	bits 19 to 26	0.212759

Test		p-value
	bits 20 to 27	0.039475
	bits 21 to 28	0.967496
	bits 22 to 29	0.742038
	bits 23 to 30	0.852440
	bits 24 to 31	0.664025
	bits 25 to 32	0.640546
PARKING	1st	0.853193
	2nd	0.659449
	3rd	0.481790
	4th	0.659449
	5th	0.767486
	6th	0.723613
	7th	0.357445
	8th	0.100530
	9th	0.374623
	10th	0.625377
K-S test for 10 p-values		0.509465
MDIST		0.906752
SPHERE	1st	0.32600
	2nd	0.82439
	3rd	0.03554
	4th	0.67480
	5th	0.64824
	6th	0.77630
	7th	0.68713
	8th	0.21617
	9th	0.13792
	10th	0.41616
	11th	0.03112
	12th	0.65695
	13th	0.79027
	14th	0.33711
	15th	0.68957
	16th	0.14503
	17th	0.98231
	18th	0.25226
	19th	0.20049
	20th	0.68244
K-S test for 20 p-values		0.247693
SQUEEZE		0.170055
OSUMS	1st	0.224604
	2nd	0.295827
	3nd	0.319191
	4nd	0.304288
	5nd	0.086728
	6nd	0.843053
	7nd	0.226564
	8nd	0.989154
	9nd	0.753418
	10nd	0.962030
K-S test for 10 p-values		0.557772
RUNS	UP 1st	0.493002
	DOWN 1st	0.729682
	UP 2nd	0.574884
	DOWN 2nd	0.289951
CRAPS	No. of wins	0.297499
	Throws/game	0.974715

Table 42: DIEHARD test results for LCG$(3512401965023503517, 0, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.918175
	bits 2 to 25	0.224224
	bits 3 to 25	0.814329
	bits 4 to 25	0.901392
	bits 5 to 25	0.899613
	bits 6 to 25	0.600892
	bits 7 to 25	0.136830
	bits 8 to 25	0.144277
	bits 9 to 25	0.796218
K-S test for 9 p-values		0.772179
OPERM	1st	0.764217
	2nd	0.357683
RANK 31×31		0.819316
RANK 32×32		0.358317
RANK 6×8	bits 1 to 8	0.564364
	bits 2 to 9	0.624702
	bits 3 to 10	0.307819
	bits 4 to 11	0.216510
	bits 5 to 12	0.663265
	bits 6 to 13	0.451514
	bits 7 to 14	0.355193
	bits 8 to 15	0.090266
	bits 9 to 16	0.036820
	bits 10 to 17	0.041455
	bits 11 to 18	0.637093
	bits 12 to 19	0.025349
	bits 13 to 20	0.795626
	bits 14 to 21	0.121542
	bits 15 to 22	0.964259
	bits 16 to 23	0.867717
	bits 17 to 24	0.725171
	bits 18 to 25	0.212474
	bits 19 to 26	0.776837
	bits 20 to 27	0.330964
	bits 21 to 28	0.361082
	bits 22 to 29	0.903910
	bits 23 to 30	0.394025
	bits 24 to 31	0.031184
	bits 25 to 32	0.269741
K-S test for 25 p-values		0.648795
BSTREAM	1st	0.53042
	2nd	0.79170
	3rd	0.54990
	4th	0.95987
	5th	0.96485
	6th	0.41916
	7th	0.10090
	8th	0.17420
	9th	0.89925
	10th	0.23277
	11th	0.42740
	12th	0.44765
	13th	0.04019

Test		p-value
	14th	0.46247
	15th	0.57200
	16th	0.02605
	17th	0.96557
	18th	0.10508
	19th	0.22851
	20th	0.44026
OPSO	bits 23 to 32	0.7322
	bits 22 to 31	0.9606
	bits 21 to 30	0.1660
	bits 20 to 29	0.1002
	bits 19 to 28	0.8360
	bits 18 to 27	0.3777
	bits 17 to 26	0.6525
	bits 16 to 25	0.2115
	bits 15 to 24	0.3179
	bits 14 to 23	0.7242
	bits 13 to 22	0.4817
	bits 12 to 21	0.7288
	bits 11 to 20	0.7777
	bits 10 to 19	0.1884
	bits 9 to 18	0.6122
	bits 8 to 17	0.4393
	bits 7 to 16	0.9679
	bits 6 to 15	0.6109
	bits 5 to 14	0.3621
	bits 4 to 13	0.6886
	bits 3 to 12	0.6109
	bits 2 to 11	0.0186
	bits 1 to 10	0.4611
OQSO	bits 28 to 32	0.8361
	bits 27 to 31	0.2067
	bits 26 to 30	0.3849
	bits 25 to 29	0.9857
	bits 24 to 28	0.4820
	bits 23 to 27	0.9718
	bits 22 to 26	0.1329
	bits 21 to 25	0.2773
	bits 20 to 24	0.3580
	bits 19 to 23	0.5158
	bits 18 to 22	0.2705
	bits 17 to 21	0.8821
	bits 16 to 20	0.6929
	bits 15 to 19	0.3492
	bits 14 to 18	0.9133
	bits 13 to 17	0.4564
	bits 12 to 16	0.5185
	bits 11 to 15	0.2307
	bits 10 to 14	0.8106
	bits 9 to 13	0.0609
	bits 8 to 12	0.5934
	bits 7 to 11	0.7288
	bits 6 to 10	0.0896

Test		p-value
	bits 5 to 9	0.2135
	bits 4 to 8	0.0290
	bits 3 to 7	0.5508
	bits 2 to 6	0.7603
	bits 1 to 5	0.6361
DNA	bits 31 to 32	0.8655
	bits 30 to 31	0.7009
	bits 29 to 30	0.7211
	bits 28 to 29	0.3287
	bits 27 to 28	0.2050
	bits 26 to 27	0.3213
	bits 25 to 26	0.5710
	bits 24 to 25	0.8661
	bits 23 to 24	0.5290
	bits 22 to 23	0.8003
	bits 21 to 22	0.8346
	bits 20 to 21	0.5998
	bits 19 to 20	0.2569
	bits 18 to 19	0.1303
	bits 17 to 18	0.3904
	bits 16 to 17	0.5055
	bits 15 to 16	0.9290
	bits 14 to 15	0.7009
	bits 13 to 14	0.7919
	bits 12 to 13	0.5617
	bits 11 to 12	0.3395
	bits 10 to 11	0.2126
	bits 9 to 10	0.0146
	bits 8 to 9	0.4167
	bits 7 to 8	0.2126
	bits 6 to 7	0.2401
	bits 5 to 6	0.0757
	bits 4 to 5	0.5008
	bits 3 to 4	0.0765
	bits 2 to 3	0.7500
	bits 1 to 2	0.8256
COUNT1S	1st	0.722610
	2nd	0.354334
COUNT1B	bits 1 to 8	0.691069
	bits 2 to 9	0.027032
	bits 3 to 10	0.630820
	bits 4 to 11	0.206829
	bits 5 to 12	0.971943
	bits 6 to 13	0.839165
	bits 7 to 14	0.892951
	bits 8 to 15	0.974180
	bits 9 to 16	0.373406
	bits 10 to 17	**0.992262**
	bits 11 to 18	0.439712
	bits 12 to 19	0.650300
	bits 13 to 20	0.796946
	bits 14 to 21	0.604809
	bits 15 to 22	0.183616
	bits 16 to 23	0.201743
	bits 17 to 24	0.582343
	bits 18 to 25	0.720871
	bits 19 to 26	0.789796

Test		p-value
	bits 20 to 27	0.089912
	bits 21 to 28	0.363575
	bits 22 to 29	0.065150
	bits 23 to 30	0.125726
	bits 24 to 31	0.356028
	bits 25 to 32	0.639261
PARKING	1st	0.781201
	2nd	0.554479
	3rd	0.261324
	4th	0.136563
	5th	0.323972
	6th	0.085365
	7th	0.781201
	8th	0.445521
	9th	0.168804
	10th	0.340551
K-S test for 10 p-values		0.651458
MDIST		0.238350
SPHERE	1st	0.73870
	2nd	0.74901
	3rd	0.33461
	4th	0.77565
	5th	0.66809
	6th	0.45308
	7th	0.58390
	8th	0.09677
	9th	0.28968
	10th	0.60912
	11th	0.05553
	12th	0.13490
	13th	0.18493
	14th	0.69821
	15th	0.05054
	16th	0.38510
	17th	0.37116
	18th	0.19396
	19th	0.73248
	20th	0.34286
K-S test for 20 p-values		0.708280
SQUEEZE		0.282916
OSUMS	1st	0.812200
	2nd	0.485152
	3nd	0.654435
	4nd	0.249923
	5nd	0.728370
	6nd	0.472731
	7nd	0.179069
	8nd	0.556552
	9nd	0.602433
	10nd	0.332094
K-S test for 10 p-values		0.429908
RUNS	UP 1st	0.199066
	DOWN 1st	0.484925
	UP 2nd	0.398951
	DOWN 2nd	0.741266
CRAPS	No. of wins	0.121663
	Throws/game	0.871119

Table 43: DIEHARD test results for LCG$(2444805353187672469, 0, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.193545
	bits 2 to 25	0.532924
	bits 3 to 25	0.268666
	bits 4 to 25	0.644471
	bits 5 to 25	0.397855
	bits 6 to 25	0.899574
	bits 7 to 25	0.386786
	bits 8 to 25	0.576710
	bits 9 to 25	0.318485
K-S test for 9 p-values		0.415225
OPERM	1st	0.353411
	2nd	0.443585
RANK 31×31		0.325080
RANK 32×32		0.556414
RANK 6×8	bits 1 to 8	0.041842
	bits 2 to 9	0.171740
	bits 3 to 10	0.089051
	bits 4 to 11	0.773874
	bits 5 to 12	0.103779
	bits 6 to 13	0.109914
	bits 7 to 14	0.208084
	bits 8 to 15	0.968900
	bits 9 to 16	0.925074
	bits 10 to 17	0.926686
	bits 11 to 18	0.915222
	bits 12 to 19	0.225424
	bits 13 to 20	0.212238
	bits 14 to 21	0.291307
	bits 15 to 22	0.790935
	bits 16 to 23	0.290217
	bits 17 to 24	0.511052
	bits 18 to 25	0.867991
	bits 19 to 26	0.278514
	bits 20 to 27	0.313323
	bits 21 to 28	0.351587
	bits 22 to 29	0.718712
	bits 23 to 30	0.094206
	bits 24 to 31	0.025524
	bits 25 to 32	0.681659
K-S test for 25 p-values		0.822926
BSTREAM	1st	0.22781
	2nd	0.52857
	3rd	0.86274
	4th	0.33847
	5th	**0.99243**
	6th	0.77032
	7th	0.42923
	8th	0.94746
	9th	0.59751
	10th	0.38662
	11th	0.12743
	12th	0.16419
	13th	0.62437

Test		p-value
	14th	0.51367
	15th	0.95589
	16th	0.20321
	17th	0.79102
	18th	0.72798
	19th	0.34964
	20th	0.35921
OPSO	bits 23 to 32	0.9105
	bits 22 to 31	0.0864
	bits 21 to 30	0.2470
	bits 20 to 29	0.2503
	bits 19 to 28	0.7694
	bits 18 to 27	0.5734
	bits 17 to 26	0.7031
	bits 16 to 25	0.8715
	bits 15 to 24	0.0356
	bits 14 to 23	0.5489
	bits 13 to 22	0.8427
	bits 12 to 21	0.4461
	bits 11 to 20	0.8525
	bits 10 to 19	0.8220
	bits 9 to 18	0.8913
	bits 8 to 17	0.6056
	bits 7 to 16	0.7446
	bits 6 to 15	0.0389
	bits 5 to 14	0.1122
	bits 4 to 13	0.7161
	bits 3 to 12	0.7947
	bits 2 to 11	0.2448
	bits 1 to 10	**0.9979**
OQSO	bits 28 to 32	0.4982
	bits 27 to 31	0.6637
	bits 26 to 30	0.5212
	bits 25 to 29	0.3927
	bits 24 to 28	0.5775
	bits 23 to 27	0.7453
	bits 22 to 26	0.1693
	bits 21 to 25	0.0992
	bits 20 to 24	0.1617
	bits 19 to 23	0.4698
	bits 18 to 22	0.6348
	bits 17 to 21	0.4032
	bits 16 to 20	0.2561
	bits 15 to 19	0.7254
	bits 14 to 18	0.8821
	bits 13 to 17	0.5454
	bits 12 to 16	0.8361
	bits 11 to 15	0.7666
	bits 10 to 14	0.7728
	bits 9 to 13	0.9615
	bits 8 to 12	0.4403
	bits 7 to 11	0.6772
	bits 6 to 10	0.0514

Test		p-value
	bits 5 to 9	0.0536
	bits 4 to 8	0.9525
	bits 3 to 7	0.0421
	bits 2 to 6	0.9288
	bits 1 to 5	0.4631
DNA	bits 31 to 32	0.7519
	bits 30 to 31	0.4352
	bits 29 to 30	0.1480
	bits 28 to 29	0.2852
	bits 27 to 28	0.7171
	bits 26 to 27	0.2724
	bits 25 to 26	0.5196
	bits 24 to 25	0.9613
	bits 23 to 24	0.7970
	bits 22 to 23	0.6280
	bits 21 to 22	0.2410
	bits 20 to 21	0.3384
	bits 19 to 20	0.3319
	bits 18 to 19	0.2733
	bits 17 to 18	0.6391
	bits 16 to 17	0.8927
	bits 15 to 16	0.4655
	bits 14 to 15	0.0911
	bits 13 to 14	0.2646
	bits 12 to 13	0.7280
	bits 11 to 12	0.6652
	bits 10 to 11	0.8382
	bits 9 to 10	0.5524
	bits 8 to 9	0.4996
	bits 7 to 8	0.0624
	bits 6 to 7	0.6968
	bits 5 to 6	0.0468
	bits 4 to 5	0.5008
	bits 3 to 4	**0.0004**
	bits 2 to 3	0.8866
	bits 1 to 2	0.8954
COUNT1S	1st	0.207042
	2nd	0.961829
COUNT1B	bits 1 to 8	0.224769
	bits 2 to 9	0.731407
	bits 3 to 10	0.507596
	bits 4 to 11	0.084269
	bits 5 to 12	0.263026
	bits 6 to 13	0.174003
	bits 7 to 14	0.938141
	bits 8 to 15	0.379658
	bits 9 to 16	0.783477
	bits 10 to 17	0.728043
	bits 11 to 18	0.754630
	bits 12 to 19	0.534358
	bits 13 to 20	0.605773
	bits 14 to 21	0.765819
	bits 15 to 22	0.885956
	bits 16 to 23	0.671126
	bits 17 to 24	0.702257
	bits 18 to 25	0.641689
	bits 19 to 26	0.602335

Test		p-value
	bits 20 to 27	0.819698
	bits 21 to 28	0.584896
	bits 22 to 29	0.642023
	bits 23 to 30	0.733239
	bits 24 to 31	0.033304
	bits 25 to 32	0.464097
PARKING	1st	**0.009936**
	2nd	0.276387
	3rd	0.463618
	4th	0.055002
	5th	0.781201
	6th	0.518210
	7th	0.853193
	8th	0.590298
	9th	0.853193
	10th	0.831196
K-S test for 10 p-values		0.343457
MDIST		0.897445
SPHERE	1st	0.34716
	2nd	0.48100
	3rd	0.43662
	4th	0.30058
	5th	0.35877
	6th	0.54640
	7th	**0.99834**
	8th	0.83490
	9th	0.50928
	10th	0.84301
	11th	0.13023
	12th	0.06790
	13th	0.16116
	14th	**0.00479**
	15th	0.08544
	16th	0.57751
	17th	0.22946
	18th	0.95542
	19th	0.08717
	20th	0.09574
K-S test for 20 p-values		0.863873
SQUEEZE		0.881511
OSUMS	1st	0.706169
	2nd	0.233125
	3nd	0.431244
	4nd	0.629350
	5nd	0.771801
	6nd	0.754542
	7nd	0.893973
	8nd	0.211153
	9nd	0.468310
	10nd	0.946623
K-S test for 10 p-values		0.550182
RUNS	UP 1st	0.772599
	DOWN 1st	0.682501
	UP 2nd	0.801633
	DOWN 2nd	0.287603
CRAPS	No. of wins	0.902801
	Throws/game	0.731621

Table 44: DIEHARD test results for LCG$(1987591058829310733, 0, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.740385
	bits 2 to 25	0.972517
	bits 3 to 25	0.894966
	bits 4 to 25	0.401502
	bits 5 to 25	0.471229
	bits 6 to 25	0.585151
	bits 7 to 25	0.566122
	bits 8 to 25	0.589979
	bits 9 to 25	0.874326
K-S test for 9 p-values		0.904624
OPERM	1st	0.638675
	2nd	0.809690
RANK 31×31		0.383637
RANK 32×32		0.983716
RANK 6×8	bits 1 to 8	0.760367
	bits 2 to 9	0.224131
	bits 3 to 10	0.288892
	bits 4 to 11	0.187880
	bits 5 to 12	0.546135
	bits 6 to 13	0.764841
	bits 7 to 14	0.958769
	bits 8 to 15	0.665375
	bits 9 to 16	0.078016
	bits 10 to 17	0.044300
	bits 11 to 18	0.337731
	bits 12 to 19	0.778755
	bits 13 to 20	0.780117
	bits 14 to 21	0.340914
	bits 15 to 22	0.439145
	bits 16 to 23	0.962073
	bits 17 to 24	0.961016
	bits 18 to 25	0.246127
	bits 19 to 26	0.971007
	bits 20 to 27	0.676890
	bits 21 to 28	0.604962
	bits 22 to 29	0.506288
	bits 23 to 30	0.339887
	bits 24 to 31	0.021473
	bits 25 to 32	0.740360
K-S test for 25 p-values		0.236210
BSTREAM	1st	0.75443
	2nd	0.85542
	3rd	0.11339
	4th	0.66097
	5th	0.31322
	6th	0.78629
	7th	0.88587
	8th	0.18027
	9th	0.26535
	10th	0.85436
	11th	0.03344
	12th	0.87802
	13th	0.09804

Test		p-value
	14th	0.12841
	15th	0.61280
	16th	0.87563
	17th	0.95718
	18th	0.29685
	19th	0.01227
	20th	0.77804
OPSO	bits 23 to 32	0.4666
	bits 22 to 31	0.2961
	bits 21 to 30	0.5585
	bits 20 to 29	0.5339
	bits 19 to 28	0.1893
	bits 18 to 27	0.9868
	bits 17 to 26	0.8656
	bits 16 to 25	0.4082
	bits 15 to 24	0.0937
	bits 14 to 23	0.2331
	bits 13 to 22	0.1686
	bits 12 to 21	0.1969
	bits 11 to 20	0.4927
	bits 10 to 19	**0.0049**
	bits 9 to 18	0.6422
	bits 8 to 17	0.9502
	bits 7 to 16	0.9303
	bits 6 to 15	0.2056
	bits 5 to 14	0.1083
	bits 4 to 13	0.9105
	bits 3 to 12	0.6813
	bits 2 to 11	0.1302
	bits 1 to 10	0.0698
OQSO	bits 28 to 32	0.7140
	bits 27 to 31	0.5050
	bits 26 to 30	0.6297
	bits 25 to 29	0.8079
	bits 24 to 28	0.2400
	bits 23 to 27	0.3707
	bits 22 to 26	0.6233
	bits 21 to 25	0.5454
	bits 20 to 24	0.2019
	bits 19 to 23	0.1650
	bits 18 to 22	0.6051
	bits 17 to 21	0.5468
	bits 16 to 20	0.5748
	bits 15 to 19	0.6462
	bits 14 to 18	0.7829
	bits 13 to 17	0.1433
	bits 12 to 16	0.9067
	bits 11 to 15	0.4389
	bits 10 to 14	0.2761
	bits 9 to 13	0.8893
	bits 8 to 12	0.7878
	bits 7 to 11	0.6538
	bits 6 to 10	0.6575

Test		p-value
	bits 5 to 9	0.8060
	bits 4 to 8	0.3184
	bits 3 to 7	0.9795
	bits 2 to 6	0.2539
	bits 1 to 5	0.0577
DNA	bits 31 to 32	0.8124
	bits 30 to 31	0.4375
	bits 29 to 30	0.6916
	bits 28 to 29	0.0684
	bits 27 to 28	0.6706
	bits 26 to 27	0.8068
	bits 25 to 26	0.2783
	bits 24 to 25	0.2221
	bits 23 to 24	0.3046
	bits 22 to 23	0.2447
	bits 21 to 22	0.4573
	bits 20 to 21	0.2230
	bits 19 to 20	0.0770
	bits 18 to 19	0.0650
	bits 17 to 18	0.6280
	bits 16 to 17	0.0930
	bits 15 to 16	0.3223
	bits 14 to 15	0.1800
	bits 13 to 14	0.2283
	bits 12 to 13	0.1862
	bits 11 to 12	0.2117
	bits 10 to 11	0.1297
	bits 9 to 10	0.0532
	bits 8 to 9	0.2392
	bits 7 to 8	0.9184
	bits 6 to 7	0.3223
	bits 5 to 6	0.8888
	bits 4 to 5	0.8432
	bits 3 to 4	0.2636
	bits 2 to 3	0.7746
	bits 1 to 2	0.1870
COUNT1S	1st	0.455242
	2nd	0.153778
COUNT1B	bits 1 to 8	0.964345
	bits 2 to 9	0.826983
	bits 3 to 10	0.140691
	bits 4 to 11	0.648583
	bits 5 to 12	0.671401
	bits 6 to 13	0.483024
	bits 7 to 14	0.580038
	bits 8 to 15	0.203769
	bits 9 to 16	0.154869
	bits 10 to 17	0.669677
	bits 11 to 18	0.445223
	bits 12 to 19	0.142196
	bits 13 to 20	0.893441
	bits 14 to 21	0.845237
	bits 15 to 22	0.837701
	bits 16 to 23	0.722837
	bits 17 to 24	0.970731
	bits 18 to 25	0.746586
	bits 19 to 26	0.749700

Test		p-value
	bits 20 to 27	0.633061
	bits 21 to 28	**0.998769**
	bits 22 to 29	0.382028
	bits 23 to 30	0.480145
	bits 24 to 31	0.209215
	bits 25 to 32	0.759778
PARKING	1st	**0.006836**
	2nd	0.481790
	3rd	0.831196
	4th	0.590298
	5th	0.572463
	6th	0.071982
	7th	0.819442
	8th	0.027568
	9th	0.340551
	10th	0.092718
K-S test for 10 p-values		0.849052
MDIST		0.701685
SPHERE	1st	0.26858
	2nd	0.18854
	3rd	0.38634
	4th	**0.99945**
	5th	0.79293
	6th	0.95899
	7th	0.05618
	8th	0.91957
	9th	0.89502
	10th	0.94405
	11th	0.79461
	12th	0.25310
	13th	0.72640
	14th	0.31612
	15th	0.21110
	16th	0.84962
	17th	0.87688
	18th	0.27824
	19th	0.56252
	20th	0.92306
K-S test for 20 p-values		0.954129
SQUEEZE		0.519053
OSUMS	1st	0.601707
	2nd	0.632279
	3nd	0.153232
	4nd	0.688688
	5nd	0.096181
	6nd	0.787407
	7nd	**0.001462**
	8nd	0.149491
	9nd	0.789830
	10nd	0.217253
K-S test for 10 p-values		0.731558
RUNS	UP 1st	0.297063
	DOWN 1st	0.776076
	UP 2nd	0.349017
	DOWN 2nd	0.026262
CRAPS	No. of wins	0.153014
	Throws/game	0.224485

Table 45: DIEHARD test results for LCG$(9219741426499971445, 1, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.720403
	bits 2 to 25	0.821367
	bits 3 to 25	0.663259
	bits 4 to 25	0.885925
	bits 5 to 25	0.088926
	bits 6 to 25	0.176140
	bits 7 to 25	0.486107
	bits 8 to 25	0.595153
	bits 9 to 25	0.373985
K-S test for 9 p-values		0.040916
OPERM	1st	0.985712
	2nd	0.273085
RANK 31×31		0.680562
RANK 32×32		0.327686
RANK 6×8	bits 1 to 8	0.359260
	bits 2 to 9	0.680324
	bits 3 to 10	0.089763
	bits 4 to 11	0.798436
	bits 5 to 12	0.363833
	bits 6 to 13	0.896677
	bits 7 to 14	0.639693
	bits 8 to 15	0.466500
	bits 9 to 16	0.396577
	bits 10 to 17	0.345761
	bits 11 to 18	0.312819
	bits 12 to 19	0.630386
	bits 13 to 20	0.415567
	bits 14 to 21	0.519402
	bits 15 to 22	0.578161
	bits 16 to 23	0.570128
	bits 17 to 24	0.411871
	bits 18 to 25	0.187655
	bits 19 to 26	0.829884
	bits 20 to 27	0.329908
	bits 21 to 28	**0.000049**
	bits 22 to 29	0.665247
	bits 23 to 30	0.248845
	bits 24 to 31	**0.998480**
	bits 25 to 32	0.966068
K-S test for 25 p-values		0.514648
BSTREAM	1st	0.21874
	2nd	0.31322
	3rd	0.11249
	4th	0.84002
	5th	0.70343
	6th	0.78900
	7th	0.35572
	8th	0.22081
	9th	0.09058
	10th	0.17905
	11th	0.50156
	12th	0.06345
	13th	0.39649

Test		p-value
	14th	0.61369
	15th	0.50529
	16th	**0.00331**
	17th	0.06432
	18th	0.74024
	19th	0.15289
	20th	0.30417
OPSO	bits 23 to 32	0.9725
	bits 22 to 31	0.1377
	bits 21 to 30	0.7888
	bits 20 to 29	0.5051
	bits 19 to 28	0.6651
	bits 18 to 27	0.7067
	bits 17 to 26	0.7102
	bits 16 to 25	0.4406
	bits 15 to 24	0.0246
	bits 14 to 23	0.1792
	bits 13 to 22	0.2514
	bits 12 to 21	0.6983
	bits 11 to 20	0.6837
	bits 10 to 19	0.5188
	bits 9 to 18	0.6664
	bits 8 to 17	0.9154
	bits 7 to 16	0.9550
	bits 6 to 15	0.5243
	bits 5 to 14	0.2066
	bits 4 to 13	0.7219
	bits 3 to 12	0.1332
	bits 2 to 11	0.0212
	bits 1 to 10	0.8247
OQSO	bits 28 to 32	0.7321
	bits 27 to 31	0.5535
	bits 26 to 30	0.1789
	bits 25 to 29	0.5629
	bits 24 to 28	0.7937
	bits 23 to 27	0.9849
	bits 22 to 26	0.8515
	bits 21 to 25	0.1250
	bits 20 to 24	0.6386
	bits 19 to 23	0.4901
	bits 18 to 22	0.5575
	bits 17 to 21	0.2155
	bits 16 to 20	0.6271
	bits 15 to 19	0.1216
	bits 14 to 18	0.4230
	bits 13 to 17	0.6259
	bits 12 to 16	0.2899
	bits 11 to 15	0.5171
	bits 10 to 14	0.0274
	bits 9 to 13	0.5414
	bits 8 to 12	0.1543
	bits 7 to 11	0.2215
	bits 6 to 10	0.4860

Test		p-value
	bits 5 to 9	0.1307
	bits 4 to 8	0.2165
	bits 3 to 7	0.1182
	bits 2 to 6	0.5854
	bits 1 to 5	0.8814
DNA	bits 31 to 32	0.3972
	bits 30 to 31	0.2579
	bits 29 to 30	0.8093
	bits 28 to 29	0.4305
	bits 27 to 28	0.4224
	bits 26 to 27	0.5020
	bits 25 to 26	0.6213
	bits 24 to 25	0.3938
	bits 23 to 24	0.4831
	bits 22 to 23	0.6706
	bits 21 to 22	0.8742
	bits 20 to 21	0.2186
	bits 19 to 20	0.6302
	bits 18 to 19	0.0232
	bits 17 to 18	0.7970
	bits 16 to 17	0.5814
	bits 15 to 16	0.5837
	bits 14 to 15	0.9236
	bits 13 to 14	0.2100
	bits 12 to 13	0.7702
	bits 11 to 12	0.6134
	bits 10 to 11	0.6577
	bits 9 to 10	0.4796
	bits 8 to 9	0.6716
	bits 7 to 8	0.8346
	bits 6 to 7	0.3848
	bits 5 to 6	0.2319
	bits 4 to 5	0.9115
	bits 3 to 4	0.8563
	bits 2 to 3	0.3724
	bits 1 to 2	0.2337
COUNT1S	1st	0.295804
	2nd	0.594562
COUNT1B	bits 1 to 8	0.711647
	bits 2 to 9	**0.009409**
	bits 3 to 10	0.601208
	bits 4 to 11	0.673514
	bits 5 to 12	0.563164
	bits 6 to 13	0.276289
	bits 7 to 14	0.915314
	bits 8 to 15	0.764624
	bits 9 to 16	0.910444
	bits 10 to 17	0.920692
	bits 11 to 18	0.622088
	bits 12 to 19	0.074680
	bits 13 to 20	0.234398
	bits 14 to 21	0.064607
	bits 15 to 22	0.412969
	bits 16 to 23	0.226956
	bits 17 to 24	0.289046
	bits 18 to 25	0.699915
	bits 19 to 26	0.699339

Test		p-value
	bits 20 to 27	0.818506
	bits 21 to 28	0.849217
	bits 22 to 29	0.265446
	bits 23 to 30	0.293536
	bits 24 to 31	0.892887
	bits 25 to 32	0.971628
PARKING	1st	0.409702
	2nd	0.205562
	3rd	0.625377
	4th	0.842447
	5th	0.738676
	6th	0.819442
	7th	0.767486
	8th	0.146807
	9th	0.192812
	10th	0.261324
K-S test for 10 p-values		0.177261
MDIST		0.479727
SPHERE	1st	0.69914
	2nd	0.14687
	3rd	0.77199
	4th	0.44222
	5th	0.46998
	6th	0.64175
	7th	0.23570
	8th	0.47784
	9th	0.27807
	10th	0.03189
	11th	0.33734
	12th	0.29425
	13th	0.92031
	14th	0.29867
	15th	0.71952
	16th	0.26455
	17th	0.36959
	18th	0.64885
	19th	0.78094
	20th	0.05433
K-S test for 20 p-values		0.484610
SQUEEZE		0.084253
OSUMS	1st	0.473127
	2nd	0.285556
	3nd	0.719631
	4nd	0.284637
	5nd	0.489931
	6nd	0.087302
	7nd	**0.003964**
	8nd	0.429749
	9nd	0.953740
	10nd	0.410872
K-S test for 10 p-values		0.570837
RUNS	UP 1st	0.067298
	DOWN 1st	0.224690
	UP 2nd	0.440729
	DOWN 2nd	0.013198
CRAPS	No. of wins	0.788498
	Throws/game	0.090510

Table 46: DIEHARD test results for LCG$(2806196910506780709, 1, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.210576
	bits 2 to 25	0.567063
	bits 3 to 25	0.395163
	bits 4 to 25	0.150031
	bits 5 to 25	0.412217
	bits 6 to 25	0.515134
	bits 7 to 25	0.398347
	bits 8 to 25	0.021044
	bits 9 to 25	0.489100
K-S test for 9 p-values		0.874991
OPERM	1st	0.313056
	2nd	0.161744
RANK 31×31		0.326191
RANK 32×32		0.376873
RANK 6×8	bits 1 to 8	0.156941
	bits 2 to 9	0.780946
	bits 3 to 10	0.851335
	bits 4 to 11	0.572384
	bits 5 to 12	0.277055
	bits 6 to 13	0.398422
	bits 7 to 14	0.438787
	bits 8 to 15	0.031188
	bits 9 to 16	0.578488
	bits 10 to 17	0.448589
	bits 11 to 18	0.201326
	bits 12 to 19	**0.000162**
	bits 13 to 20	0.429916
	bits 14 to 21	0.834461
	bits 15 to 22	**0.003385**
	bits 16 to 23	0.026825
	bits 17 to 24	0.607648
	bits 18 to 25	0.898203
	bits 19 to 26	0.447794
	bits 20 to 27	0.460637
	bits 21 to 28	0.651161
	bits 22 to 29	0.776687
	bits 23 to 30	0.126257
	bits 24 to 31	0.416978
	bits 25 to 32	0.581032
K-S test for 25 p-values		0.820042
BSTREAM	1st	0.54064
	2nd	0.66948
	3rd	0.13438
	4th	0.32238
	5th	0.65583
	6th	0.91256
	7th	0.65840
	8th	0.26842
	9th	0.56282
	10th	0.15510
	11th	0.06641
	12th	0.82480
	13th	0.45876

Test		p-value
	14th	0.17420
	15th	0.21599
	16th	0.15179
	17th	0.19090
	18th	0.29363
	19th	0.39199
	20th	0.28010
OPSO	bits 23 to 32	0.9077
	bits 22 to 31	0.4693
	bits 21 to 30	0.3216
	bits 20 to 29	0.0827
	bits 19 to 28	0.6461
	bits 18 to 27	0.2648
	bits 17 to 26	0.0645
	bits 16 to 25	0.1109
	bits 15 to 24	0.1362
	bits 14 to 23	0.7683
	bits 13 to 22	0.1014
	bits 12 to 21	0.1651
	bits 11 to 20	0.0562
	bits 10 to 19	0.3416
	bits 9 to 18	0.1231
	bits 8 to 17	0.4109
	bits 7 to 16	0.4447
	bits 6 to 15	0.9370
	bits 5 to 14	0.7797
	bits 4 to 13	0.8343
	bits 3 to 12	0.2373
	bits 2 to 11	0.1686
	bits 1 to 10	0.1584
OQSO	bits 28 to 32	0.7635
	bits 27 to 31	0.3875
	bits 26 to 30	**0.9967**
	bits 25 to 29	0.5360
	bits 24 to 28	0.0643
	bits 23 to 27	0.6513
	bits 22 to 26	0.3784
	bits 21 to 25	0.2594
	bits 20 to 24	0.4309
	bits 19 to 23	0.5117
	bits 18 to 22	0.6525
	bits 17 to 21	0.4124
	bits 16 to 20	0.7635
	bits 15 to 19	0.7937
	bits 14 to 18	0.6625
	bits 13 to 17	0.2605
	bits 12 to 16	0.2969
	bits 11 to 15	0.4443
	bits 10 to 14	0.5615
	bits 9 to 13	0.9112
	bits 8 to 12	0.5198
	bits 7 to 11	0.4230
	bits 6 to 10	0.2175

Test		p-value
	bits 5 to 9	0.5090
	bits 4 to 8	0.3862
	bits 3 to 7	0.1860
	bits 2 to 6	0.0566
	bits 1 to 5	0.0400
DNA	bits 31 to 32	0.1754
	bits 30 to 31	0.7584
	bits 29 to 30	0.9544
	bits 28 to 29	0.8346
	bits 27 to 28	0.5826
	bits 26 to 27	0.1480
	bits 25 to 26	0.5594
	bits 24 to 25	0.1584
	bits 23 to 24	0.3046
	bits 22 to 23	0.6522
	bits 21 to 22	0.2283
	bits 20 to 21	0.3949
	bits 19 to 20	0.2092
	bits 18 to 19	0.6770
	bits 17 to 18	0.5791
	bits 16 to 17	0.0483
	bits 15 to 16	0.8003
	bits 14 to 15	0.2025
	bits 13 to 14	0.9244
	bits 12 to 13	0.2933
	bits 11 to 12	0.8076
	bits 10 to 11	0.9032
	bits 9 to 10	0.4445
	bits 8 to 9	0.0572
	bits 7 to 8	**0.0020**
	bits 6 to 7	0.5629
	bits 5 to 6	0.4422
	bits 4 to 5	0.1136
	bits 3 to 4	0.4468
	bits 2 to 3	0.9574
	bits 1 to 2	0.1967
COUNT1S	1st	0.389222
	2nd	0.142584
COUNT1B	bits 1 to 8	0.987528
	bits 2 to 9	0.033604
	bits 3 to 10	0.122700
	bits 4 to 11	0.077063
	bits 5 to 12	0.284149
	bits 6 to 13	0.133220
	bits 7 to 14	0.883507
	bits 8 to 15	0.837384
	bits 9 to 16	0.290986
	bits 10 to 17	0.122840
	bits 11 to 18	0.448345
	bits 12 to 19	0.894082
	bits 13 to 20	0.937465
	bits 14 to 21	0.082221
	bits 15 to 22	0.984670
	bits 16 to 23	0.647900
	bits 17 to 24	0.745915
	bits 18 to 25	0.654630
	bits 19 to 26	0.519226

Test		p-value
	bits 20 to 27	0.384261
	bits 21 to 28	0.519709
	bits 22 to 29	0.636321
	bits 23 to 30	0.450539
	bits 24 to 31	0.100168
	bits 25 to 32	0.527464
PARKING	1st	0.753306
	2nd	0.045562
	3rd	0.625377
	4th	0.323972
	5th	0.445521
	6th	0.071982
	7th	0.192812
	8th	0.723613
	9th	0.117571
	10th	0.232514
K-S test for 10 p-values		0.816693
MDIST		0.550287
SPHERE	1st	0.22300
	2nd	0.97539
	3rd	0.49749
	4th	0.18686
	5th	0.75159
	6th	0.52404
	7th	0.25847
	8th	0.30720
	9th	0.75467
	10th	0.48761
	11th	0.21015
	12th	0.87452
	13th	0.29798
	14th	**0.00117**
	15th	0.30458
	16th	0.07232
	17th	0.38712
	18th	0.88621
	19th	0.48029
	20th	0.58626
K-S test for 20 p-values		0.385851
SQUEEZE		**0.991716**
OSUMS	1st	0.767464
	2nd	0.050108
	3nd	0.939012
	4nd	0.141600
	5nd	**0.006633**
	6nd	0.625941
	7nd	0.257937
	8nd	0.657818
	9nd	0.843215
	10nd	**0.004834**
K-S test for 10 p-values		0.883658
RUNS	UP 1st	0.528307
	DOWN 1st	0.514689
	UP 2nd	0.636428
	DOWN 2nd	0.909636
CRAPS	No. of wins	0.911708
	Throws/game	0.665904

Table 47: DIEHARD test results for LCG$(3249286849523012805, 1, 2^{63})$

Test		p-value
BDAY	bits 1 to 24	0.556632
	bits 2 to 25	0.950086
	bits 3 to 25	0.301639
	bits 4 to 25	0.868120
	bits 5 to 25	0.506750
	bits 6 to 25	0.622538
	bits 7 to 25	0.821106
	bits 8 to 25	0.508987
	bits 9 to 25	0.015290
K-S test for 9 p-values		0.354311
OPERM	1st	0.361794
	2nd	0.164908
RANK 31×31		0.344905
RANK 32×32		0.485728
RANK 6×8	bits 1 to 8	0.961494
	bits 2 to 9	0.605221
	bits 3 to 10	0.668058
	bits 4 to 11	0.964615
	bits 5 to 12	0.749401
	bits 6 to 13	0.059343
	bits 7 to 14	0.421747
	bits 8 to 15	0.608381
	bits 9 to 16	0.224066
	bits 10 to 17	0.569709
	bits 11 to 18	0.988024
	bits 12 to 19	0.305853
	bits 13 to 20	0.681299
	bits 14 to 21	0.197022
	bits 15 to 22	0.149847
	bits 16 to 23	0.870398
	bits 17 to 24	0.327733
	bits 18 to 25	0.313729
	bits 19 to 26	0.715364
	bits 20 to 27	0.301268
	bits 21 to 28	0.276601
	bits 22 to 29	0.511967
	bits 23 to 30	0.286821
	bits 24 to 31	0.270652
	bits 25 to 32	**0.001605**
K-S test for 25 p-values		0.287553
BSTREAM	1st	0.39199
	2nd	0.83252
	3rd	0.12597
	4th	0.20854
	5th	0.67117
	6th	0.28405
	7th	0.47454
	8th	0.47641
	9th	0.88268
	10th	0.76027
	11th	0.93400
	12th	0.22150
	13th	0.93340

Test		p-value
	14th	0.51367
	15th	0.25324
	16th	0.27151
	17th	0.33335
	18th	0.13489
	19th	0.77734
	20th	0.73107
OPSO	bits 23 to 32	0.7797
	bits 22 to 31	0.2278
	bits 21 to 30	0.8685
	bits 20 to 29	0.2514
	bits 19 to 28	0.3909
	bits 18 to 27	0.0966
	bits 17 to 26	0.7908
	bits 16 to 25	0.1643
	bits 15 to 24	0.6911
	bits 14 to 23	0.5653
	bits 13 to 22	0.6739
	bits 12 to 21	0.8394
	bits 11 to 20	0.2648
	bits 10 to 19	0.9502
	bits 9 to 18	0.4257
	bits 8 to 17	0.3021
	bits 7 to 16	0.5312
	bits 6 to 15	0.8611
	bits 5 to 14	**0.9932**
	bits 4 to 13	0.1494
	bits 3 to 12	0.6959
	bits 2 to 11	0.0155
	bits 1 to 10	0.7019
OQSO	bits 28 to 32	0.3784
	bits 27 to 31	0.7697
	bits 26 to 30	0.1365
	bits 25 to 29	0.2957
	bits 24 to 28	0.4874
	bits 23 to 27	0.6588
	bits 22 to 26	0.6808
	bits 21 to 25	0.9709
	bits 20 to 24	0.3992
	bits 19 to 23	0.1824
	bits 18 to 22	0.1135
	bits 17 to 21	0.0201
	bits 16 to 20	0.3269
	bits 15 to 19	0.0424
	bits 14 to 18	0.8974
	bits 13 to 17	**0.9974**
	bits 12 to 16	0.1293
	bits 11 to 15	0.5185
	bits 10 to 14	0.9428
	bits 9 to 13	0.9878
	bits 8 to 12	0.5454
	bits 7 to 11	0.2165
	bits 6 to 10	0.5077

Test		p-value
	bits 5 to 9	0.1149
	bits 4 to 8	0.1278
	bits 3 to 7	0.0614
	bits 2 to 6	0.7779
	bits 1 to 5	0.1789
DNA	bits 31 to 32	0.7481
	bits 30 to 31	0.8019
	bits 29 to 30	0.0361
	bits 28 to 29	0.6313
	bits 27 to 28	0.1649
	bits 26 to 27	0.2872
	bits 25 to 26	0.7782
	bits 24 to 25	0.0970
	bits 23 to 24	0.5768
	bits 22 to 23	0.5314
	bits 21 to 22	0.3881
	bits 20 to 21	0.8980
	bits 19 to 20	0.9086
	bits 18 to 19	0.9282
	bits 17 to 18	0.0210
	bits 16 to 17	0.8397
	bits 15 to 16	0.3680
	bits 14 to 15	0.4006
	bits 13 to 14	0.0804
	bits 12 to 13	0.1328
	bits 11 to 12	0.8905
	bits 10 to 11	0.7657
	bits 9 to 10	0.4831
	bits 8 to 9	0.8616
	bits 7 to 8	0.4749
	bits 6 to 7	0.5617
	bits 5 to 6	0.7151
	bits 4 to 5	0.5524
	bits 3 to 4	0.6302
	bits 2 to 3	0.1613
	bits 1 to 2	0.0980
COUNT1S	1st	0.803964
	2nd	0.473314
COUNT1B	bits 1 to 8	0.314205
	bits 2 to 9	0.271624
	bits 3 to 10	0.598896
	bits 4 to 11	0.421438
	bits 5 to 12	0.113834
	bits 6 to 13	0.118614
	bits 7 to 14	0.343508
	bits 8 to 15	0.619938
	bits 9 to 16	0.852324
	bits 10 to 17	0.232142
	bits 11 to 18	0.968922
	bits 12 to 19	0.303762
	bits 13 to 20	0.407089
	bits 14 to 21	0.115875
	bits 15 to 22	0.915336
	bits 16 to 23	0.026976
	bits 17 to 24	0.417301
	bits 18 to 25	0.599062
	bits 19 to 26	0.566468

Test		p-value
	bits 20 to 27	0.093004
	bits 21 to 28	0.137280
	bits 22 to 29	0.077813
	bits 23 to 30	0.552101
	bits 24 to 31	0.743590
	bits 25 to 32	0.908625
PARKING	1st	0.554479
	2nd	0.276387
	3rd	0.463618
	4th	0.914635
	5th	0.819442
	6th	0.218799
	7th	0.276387
	8th	0.590298
	9th	0.842447
	10th	0.642555
K-S test for 10 p-values		0.300768
MDIST		0.963428
SPHERE	1st	0.37958
	2nd	0.39040
	3rd	0.43940
	4th	0.08448
	5th	0.96621
	6th	0.35165
	7th	**0.00055**
	8th	0.64554
	9th	0.09126
	10th	0.06005
	11th	0.10416
	12th	0.76144
	13th	0.53002
	14th	0.95829
	15th	0.29646
	16th	0.85687
	17th	0.29163
	18th	0.65482
	19th	0.14456
	20th	0.29321
K-S test for 20 p-values		0.763644
SQUEEZE		0.921072
OSUMS	1st	0.448758
	2nd	0.787182
	3nd	0.931507
	4nd	0.787418
	5nd	0.157620
	6nd	0.652882
	7nd	0.636972
	8nd	0.153021
	9nd	0.676151
	10nd	0.261674
K-S test for 10 p-values		0.207865
RUNS	UP 1st	0.830472
	DOWN 1st	**0.008008**
	UP 2nd	0.675384
	DOWN 2nd	0.025304
CRAPS	No. of wins	0.699833
	Throws/game	0.610991

5 Conclusion

We summarized the principle and features of LCGs that are frequently used in particle-transport Monte Carlo methods and tests used to investigate the quality of the LCGs. We also performed the spectral test, Knuth's standard tests and Marsaglia's DIEHARD tests for the MCNP RNG, 63-bit LCGs extended from the MCNP RNG and 63-bit LCGs proposed by L'Ecuyer.

The MCNP RNG fails the OPSO, OQSO and DNA tests in the DIEHARD test suite, whereas it passes the spectral test, the standard tests and other tests in DIEHARD. However less significant bits fail the tests and thus it does not matter in the practical use.

The 63-bit LCGs extended from the MCNP RNG fail the spectral test, whereas they pass the spectral and DIEHARD tests. We have found that we cannot simply extend the current MCNP RNG to a 63-bit LCG.

L'Ecyer's 63-bit LCGs pass all the tests and their multipliers are excellent judging from the spectral test. Therefore, it is considered that they are the most promising LCGs for the next version of the RNG package.

References

[1] D. H. Lehmer, "Mathematical methods in large-scale computing units," *Proc. of the Second Symp. on Large Scale Digital Computing Machinery*, Harvard University Press, Cambridge, Massachusetts, pp.141-146 (1949).; Ann. Comp. Lab. Harvard University, **26** (1951).

[2] P. L'Ecuyer, "Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure," *Math. Comp.*, **68**, 249-260 (1999).

[3] D. E. Knuth, "The Art of Computer Programming, Vol.2: Seminumerical Algorithms," 3rd edition, Addison Wesley Longman (1998).

[4] G. Marsaglia, Diehard software package.
http://stat.fsu.edu/~geo/diehard.html

[5] R. R. Coveyou and R. D. MacPherson, "Fourier analysis of uniform random number generators," *J. Assoc. Comp. Mach.*, **14**, pp. 100–119 (1967).

[6] U. Dieter, "How to calculate shortest vectors in a lattice," *Math. Comput.*, **29**, pp. 827–833 (1975).

[7] D. E. Knuth, "The Art of Computer Programming, Vol.2: Seminumerical Algorithms," 2nd edition, Addison Wesley (1981).

[8] T. R. Hopkins, "A Revised Algorithm for the Spectral Test," *Applied Statistics*, **32**, pp. 328–335.

[9] G. Marsaglia, "Random Numbers Fall Mainly in the Planes," *Proc. National Academy of Sciences, U.S.A.*, **61**, pp.25-28 (1968).

[10] J. W. S. Cassels, "Introduction to the Geometry of Numbers," Springer (1959); Reprint of the 1971 edition (1997).

[11] G. S. Fishman and L. R. Moore, "An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$," *SIAM J. Sci. and Statist. Comput.*, **7**, pp. 129–136 (1986).

[12] G. S. Fishman, "Multiplicative Congruential Random Number Generators with Modulus 2^β: An Exhaustive Analysis for $\beta = 32$ and a Partial Analysis for $\beta = 48$," *Math. Comp.*, **54**, 331-344 (1990).

[13] P. L'Ecuyer, "Efficient and Portable Combined Random Number Generators," *Comm. ACM*, **31**, 742 (1988).

[14] Free Software Foundation, http://www.gnu.org/software/bc/bc.html

[15] G. Marsaglia, "The structure of linear congruential sequences," in *Applications of Number Theory to Numerical Analysis*, S.K. Zaremba ed., Academice Press, pp. 249–285 (1972).

[16] G. S. Fishman, "Monte Carlo, concept, Algorithm, and Applications," Springer (1995).

[17] M. Mascagni and A. Srinivasan, SPRNG: a scalable library for pseudorandom number generation. http://sprng.cs.fsu.edu/

[18] J. E. Gentle, "Random Number Generation and Monte Carlo Methods," Springer (1998).

[19] L'Ecuyer, "Random Number Generation", Chapter 4 of the Handbook on Simulation, Jerry Banks Ed., Wiley, pp.93-137 (1998).

[20] Z. W. Birnbaum and F. H. Tingey, "One-sided Confidence Contours for Probability Distribution Functions," *Annals Math. Stat.*, **22**, pp.592-596 (1951).

[21] W. H. Press, "Numerical Recipes in C, The Art of Scientific Computing Second Edition," CAMBRIDGE UNIVERSITY PRESS, Chapter14 (1992).

[22] I. Vattulainen, et. al., "A comparative study of some pseudorandom number generators," *Comp. Phys. Comm.*, **86**, 209 (1995).

[23] M. Mascagni and A. Srinivasan, "Parameterizing Parallel Multiplicative Lagged-Fibonacci Generators," submitted to *Parallel Computing*.

[24] G. Marsaglia,"A Current View of Random Number Generators," *Proc. of Computer Science and Statistics: 16th Symposium on the Interface, Atlanta, 1984* (1984).