J. KSIAM Vol.12, No.2, 109–118, 2008

A NEW VERSION OF FIRST RETURN TIME TEST OF PSEUDORANDOMNESS

DONG HAN KIM

DEPARTMENT OF MATHEMATICS, THE UNIVERSITY OF SUWON, HWASEONG 445-743, KOREA.
E-mail address: kimdh@suwon.ac.kr.

ABSTRACT. We present a new version of the first return time test for pseudorandomness. Let R_n be the first return time of initial n-block with overlapping. An algorithm to calculate the probability distribution of the first return time R_n for each starting block is presented and used to test pseudorandom number generators. The standard Z-test for $\log R_n$ is applied to test the pseudorandom number generators.

1. INTRODUCTION

We introduce a new version of testing pseudorandom number generators (PRNGs) based on the first return time of the initial n-block for some fixed length n in a randomly generated binary sequence. The first return time is closely related to entropy, which plays a key role in the information theory. Entropy is defined as the limit of $-\frac{1}{n}\sum_i p_i \log p_i$ as n goes to infinity where the p_i's are the possibility of each block of length n appears in the source. Entropy measures the amount of randomness and the maximum compression rate.

For each binary sequence $x = (x_1, x_2, \ldots)$, define the first return time (recurrence time) by

$$R_n(x) = \min\{j \geq 1 : x_1 \ldots x_n = x_{j+1} \ldots x_{j+n}\},$$

To study the data compression algorithm like the Lempel-Ziv code[19], Wyner and Ziv showed that $\frac{1}{n}\log R_n(x)$ converges to entropy in measure[16]. Since then there have been many works on the relation between entropy and the first return time (e.g. [7], [8], [11], [18]).

Define the waiting time by

$$W_n(x, y) = W_n(x_1^n, y) = \min\{j \geq 1 : x_1 \ldots x_n = y_j \ldots y_{j+n-1}\}.$$

Wyner and Ziv showed that $\frac{1}{n}\log W_n(x, y)$ converges to entropy for Markov chains[16]. Generally $\frac{1}{n}\log W_n(x, y)$ converges to the entropy for weakly Bernoulli processes, but the convergence does not hold in general ergodic processes[15].

2000 *Mathematics Subject Classification.* 65C10.

Key words and phrases. The first return time, pseudorandom number generator, probability density, entropy, randomness.

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-331-C00016).

Maurer [10] presented the nonoverlapping first return time algorithm in testing PRNGs, His algorithm corresponds to the nonoverlapping first return time:

$$R_{(n)}(x) = \min\{j \ge 1 : x_1 \dots x_n = x_{jn+1} \dots x_{jn+n}\}.$$

Put $v(r) = r\sum_{i=1}^{\infty}(1-r)^{i-1}\log_2 i$, and $w(r) = r\sum_{i=1}^{\infty}(1-r)^{i-1}(\log_2 i)^2$. Since $\Pr(B) = 2^{-n}$, $E[\log R_{(n)}] = v(2^{-n})$ and $E[(\log_2 R_{(n)})^2] = w(2^{-n})$. Then

$$\lim_{r\to 0+}[v(r) + \log r] = -\gamma/\ln 2 = -0.832\cdots \equiv C,$$

where $\gamma = \lim_{n\to\infty}(\sum_{i=1}^{n}(1/i) - \ln n)$ is Euler's constant. Similarly,

$$\lim_{r\to 0+}[w(r) - (\log r)^2 + 2C\log r] = 4.11\cdots \equiv D.$$

So the expectation of $\log R_{(n)}$ is $n + C$ and the variation is $D - C^2$. See [2] and [4] for related results.

An overlapping algorithm of the first return time test of randomness is considered by Choe and the author[3]. They studied the return time which does not allows the overlapping between the initial block and the first recurrent block:

$$R'_n(x) \equiv \min\{j \ge n : x_1 \dots x_n = x_{j+1} \dots x_{j+n}\}.$$

Though the nonoverlapping algorithm is relatively easier to analyze, but overlapping method is more natural and efficient since nonoverlapping algorithm requires n times more sample random digits to be applied. The first return time test using $R'_n(x)$ can be regards as the waiting time test of $W_n(x_1^n, x_{n+1}^\infty)$. In this article we consider the first return time $R_n(x)$ for testing pseudorandomness. Kac's lemma[5] which states that $E[R_n|x_1 \dots x_n = B] = 1/\Pr(B)$ and the convergence of $\log R_n/n$ to entropy make R_n be more natural to consider than W_n for the randomness test though the methods in this article are not very different from that of using R'_n in [3]. For a related result of testing PRNGs using the first return time on the cyclic group, see [6].

In Section 2 we develop a formula for computing $\Pr(R_n = i|x_1 \cdots x_n = B)$ exactly. First, we classify blocks with the same distribution of R_n. Next, we use two sequences $r_k(B)$ and $s_k(B)$ which have the information on the probability distribution of R_n for $x_1 \cdots x_n = B$ and find the recurrence relation among them to compute $\Pr(R_n = i|x_1 \cdots x_n = B)$ for each i. In Section 3 we apply the standard Z-tests for $\log R_n/n$. For each block of length 14 we compute the expectation and the standard deviation of $\log R_n/n$ and the deviation of the experimental data from the theoretical prediction is used to test PRNG's. Unlike the return time test in [3], we calculate the Z-values for each starting blocks, which enable us to have more sharp test result.

2. THE PROBABILITY DISTRIBUTION OF THE FIRST RETURN TIME

A *block* is a finite sequence of elements of alphabet $\mathcal{A} = \{0, 1\}$ and an n-*block* is a block of length n. For an n-block $B = b_1 b_2 \cdots b_n \in \mathcal{A}^n$, we write $B_i^j = b_i b_{i+1} \cdots b_j, 1 \le i \le j \le n$. Throughout the paper $\mathcal{A} = \{0, 1\}$.

TABLE 1. The Expectation of R_n and $\log R_n$

Block	$\overline{\Lambda}(B)$	$\Lambda(B)$	E[R_n]	E[log R_n]	Var[log R_n]
00000000	1,2,3,4,5,6,7	1	256	4.122127	18.37019
00000001	$\emptyset$	$\emptyset$	256	7.299403	2.441935
00000010	7	7	256	7.273498	2.589157
00000100	6,7	6,7	256	7.219351	2.905512
00001000	5,6,7	5,6,7	256	7.106875	3.576236
00010001	4	4	256	7.055111	3.986235
00100001	5	5	256	7.183896	3.147559
00100010	4,7	4,7	256	7.031221	4.110117
00100100	3,6,7	3,7	256	6.717126	6.102838
01000001	6	6	256	7.244771	2.763759
01000010	5,7	5,7	256	7.158986	3.283393
01001001	3,6	3	256	6.738698	6.005312
01010101	2,4,6	2	256	6.015615	10.32028

Since the distribution of return time is different from block to block. We classify the blocks to each set of blocks have the same return time distribution.

Definition 2.1. Let B be an n-block.

$$\overline{\Lambda}(B) = \{m : B_{m+1}^n = B_1^{n-m},\ 1 \le m < n\},$$
$$\Lambda(B) = \{m : m \in \overline{\Lambda}(B),\ i \nmid m \text{ for any } i \in \overline{\Lambda}(B)\}.$$

Table 2 shows $\overline{\Lambda}(B)$ and $\Lambda(B)$ for some 8-blocks. For more of the definition of $\Lambda(B)$, see [3].

Lemma 2.2 ([3], Lemma 3)**.** *Let B be an n-block.*
(i) If $\lambda \in \overline{\Lambda}(B)$ and $\lambda < m < n$, then $\lambda \in \overline{\Lambda}(B_1^m)$.
(ii) If $B = B_{m+1}^n B_1^m$ for some $1 \le m < n$, then there is $\lambda \in \Lambda(B)$ such that λ divides n and m.
(iii) If there is $\lambda \in \Lambda(B)$ with $\lambda \le n/2$, then each $\lambda' \in \Lambda(B)$, $\lambda' \ne \lambda$, satisfies $\lambda' > n - \lambda$.

Definition 2.3. Let B be an n-block. For each $k \ge n$ denote $\mathcal{F}(B,k)$ by the set of k-blocks of

$$\mathcal{F}(B,k) = \{C : C_1^n = B, C_{i+1}^{i+n} \ne B \text{ for any } i \ge 1\},$$

and for $k \ge 1$ let $\mathcal{S}(B,k)$ be the set of k-blocks defined by

$$\mathcal{S}(B,k) = \{C : (CB)_1^n = B, (CB)_{i+1}^{i+n} \ne B \text{ for any } i, 1 \le i < k\}.$$

Clearly for $k \ge n$ we have $\mathcal{S}(B,k) \subset \mathcal{F}(B,k)$. Note that

$$x_1^k \in \mathcal{F}(B,k) \ \text{ if and only if } \ x_1^n = B, \text{ and } R_n(x) > k - n, \qquad k \ge n \quad (1)$$
$$x_1^k \in \mathcal{S}(B,k) \text{ and } x_{k+1}^{k+n} = B \ \text{ if and only if } \ x_1^n = B \text{ and } R_n(x) = k, \qquad k \ge 1. \quad (2)$$

The following shows the relation between $\mathcal{F}(B,k)$ and $\mathcal{S}(B,k)$.

Lemma 2.4. *Let B be an n-block. (i) For $k > n$ we have*

$$\mathcal{F}(B,k) \cup \{CB : C \in \mathcal{S}(B,k-n)\} = \{C \in \mathcal{A}^k : C_1^{k-1} \in \mathcal{F}(B,k-1)\},$$

where the union is disjoint. (ii) For $k \geq n$ we have

$$\mathcal{F}(B,k) \setminus \mathcal{S}(B,k) = \bigcup_{\lambda \in \overline{\Lambda}(B)} \{CB_1^\lambda : C \in \mathcal{S}(B,k-\lambda)\},$$

where the unions are disjoint.

Proof. (i) By the definition of $\mathcal{F}(B,k)$ and $\mathcal{S}(B,k)$, it is clear that

$$\mathcal{F}(B,k) \cup \{CB : C \in \mathcal{S}(B,k-n)\} \subset \{C \in \mathcal{A}^k : C_1^{k-1} \in \mathcal{F}(B,k-1)\}.$$

Let C be an k-block with $C_1^{k-1} \in \mathcal{F}(B,k-1)$ Then either $C_{k-n+1}^k = B$ or $C_{k-n+1}^k \neq B$. If $C_{k-n+1}^k = B$, then $C_1^{k-n} \in \mathcal{S}(B,k)$ and $C = C_1^{k-n}B$. When $C_{k-n+1}^k \neq B$, $C \in \mathcal{F}(B,k)$.

(ii) Take a k-block $C \in \mathcal{F}(B,k) \setminus \mathcal{S}(B,k)$. Then for some s with $0 < s < n$ we have $(CB)_{k-s+1}^{k-s+n} = B$ i.e., $s \in \overline{\Lambda}(B)$ and $C_{k-s+1}^k = B_1^s$. If we put λ as the largest number of such s's, then we have $C_1^{k-\lambda} \in \mathcal{S}(B,k-\lambda)$. Hence we have

$$\mathcal{F}(B,k) \setminus \mathcal{S}(B,k) \subset \bigcup_{\lambda \in \overline{\Lambda}(B)} \{CB_1^\lambda : C \in \mathcal{S}(B,k-\lambda)\}.$$

From the definition of $\mathcal{F}(B,k)$ and $\mathcal{S}(B,k)$, we have

$$\mathcal{F}(B,k) \setminus \mathcal{S}(B,k) = \bigcup_{\lambda \in \overline{\Lambda}(B)} \{CB_1^\lambda : C \in \mathcal{S}(B,k-\lambda)\}.$$

Now we prove the disjointness of the union: Suppose that there exist $\lambda, \lambda' \in \overline{\Lambda}(B)$, $\lambda < \lambda'$ such that $CB_1^\lambda = C'B_1^{\lambda'}$ for some $C \in \mathcal{S}(B,k-\lambda)$ and $C' \in \mathcal{S}(B,k-\lambda')$. Then we have $B_1^\lambda = B_{\lambda'-\lambda+1}^{\lambda'}$. Note that Lemma 2.2(i) implies $\lambda \in \overline{\Lambda}(B_1^{\lambda'})$ and $B_{\lambda+1}^{\lambda'} = B_1^{\lambda'-\lambda}$. Hence we have

$$B_1^{\lambda'} = B_1^\lambda B_{\lambda+1}^{\lambda'} = B_{\lambda'-\lambda+1}^{\lambda'} B_{\lambda+1}^{\lambda'} = B_{\lambda'-\lambda+1}^{\lambda'} B_1^{\lambda'-\lambda},$$

and from Lemma 2.2(ii) we have $\lambda_0 \in \Lambda(B)$ such that $\lambda = \ell\lambda_0$ and $\lambda' = \ell'\lambda_0$ for some positive integers ℓ and ℓ' with $\ell < \ell'$. Hence we have $C = C'B_1^{\lambda_0} \cdots B_1^{\lambda_0}$ and this contradicts $C \in \mathcal{S}(B,k-\lambda)$. □

Definition 2.5. Define $r_k(B)$ and $s_k(B)$ by

$$r_k(B) = \Pr(x_1 \cdots x_k \in \mathcal{F}(B,k)), \qquad k \geq n,$$
$$s_k(B) = \Pr(x_1 \cdots x_k \in \mathcal{S}(B,k)), \qquad k \geq 1.$$

Then we have (1) implies that

$$\Pr(x_1^n = B,\ R_n(x) > k-n) = r_k(B) \text{ for } k \geq n$$

and (2) yields

$$\Pr(x_1^n = B,\ R_n(x) = k) = \Pr(B)s_k(B) \text{ for } k \geq 1.$$

Now we can calculate the distribution of the first return time by the following theorem, which is directly obtained by Lemma 2.4.

Theorem 2.6. *For i.i.d. processes, if $k > n$, we have*

$$r_k(B) = r_{k-1}(B) - \Pr(B)s_{k-n}(B).$$

For $k \geq n$

$$s_k(B) = r_k(B) - \sum_{\lambda \in \overline{\Lambda}(B)} \Pr(B_1^\lambda) s_{k-\lambda}(B).$$

And for initial values we have

$$r_n(B) = \Pr(B) \ \textit{and} \ s_i(B) = \begin{cases} 0 & \textit{if } i < n, i \notin \Lambda(B), \\ \Pr(B_1^i) & \textit{if } i \in \Lambda(B). \end{cases}$$

Proof. The first recurrence relation is implied (i) and The second recurrence relation is directly obtained by Lemma 2.4 (ii).

Since $\mathcal{F}(B,n) = \{B\}$, we have $r_n(B) = \Pr(B)$. When $i < n$, each $C \in \mathcal{S}(B,n)$ should satisfies that $C = B_1^i$ and $(CB)_1^n = B_1^i B_1^{n-i} = B$, which implies that $i \notin \Lambda(B)$. Therefore, if $i < n$ and $i \notin \Lambda(B)$, then an $\mathcal{S}(B,i) = \emptyset$ or $s_i = 0$ and if $i \in \Lambda(B)$, then $\mathcal{S}(B,i) = \{B_1^i\}$ or $s_i = \Pr(B_1^i)$. □

For random binary sequences we have the followings:

$$r_k(B) = r_{k-1}(B) - 2^{-n} s_{k-n}(B), \quad k > n.$$

$$s_k(B) = r_k(B) - \sum_{\lambda \in \overline{\Lambda}(B)} 2^{-\lambda} s_{k-\lambda}(B), \quad k \geq n$$

with initial values

$$r_n(B) = 2^{-n} \ \text{and} \ s_i(B) = \begin{cases} 0 & \text{if } i < n, i \notin \Lambda(B), \\ 2^{-i} & \text{if } i \in \Lambda(B). \end{cases}$$

For $(1/2, 1/2)$ i.i.d. processes the sequence $\{r_k(B)\}$ is same for the blocks with the same $\Lambda(B)$. Thus we classify all the n-blocks using $\Lambda(B)$ and compute s_k for each block B from different classes. The computation of $s_k(B)$ for every n-block B is necessary for the application in later sections and it is done recursively on computers.

TABLE 2. The Z-test for $n = 14$ and sample size = 100,000

Generator	number of blocks such that				Mean	Variance
	$Z < -2.57$	$Z < -1.96$	$Z > 1.96$	$Z > 2.57$		
Randu	6667	6737	8923	8744	4.99	799.97
ANSI	2110	2639	7394	6024	1.09	12.16
MS	2163	2692	7364	6001	1.09	11.90
Fishman	23	206	201	28	-0.01	0.78
ICG	69	398	404	81	0.00	1.00
Ran0	25	229	204	30	0.00	0.77
Ran1	31	243	242	27	0.00	0.79
Ran2	95	434	401	79	-0.01	1.02
Ran3	73	417	415	79	0.01	1.00
F90	93	437	421	101	0.01	1.03

3. TEST FOR PSEUDORANDOM NUMBER GENERATORS

We apply Z-test for $\log R_n$ to test PRNGs given in Section 4. We calculate for each n-block B the expectations and the standard deviations for $\log R_n$ numerically by computer using the values $s_k(B)$ from Theorem 2.6.

First, we construct a long binary sequence $x = (x_1 x_2 \dots)$ by juxtaposing binary numbers of the random bits from pseudorandom number generators. Define $L(B)$ by

$$L(B) = \{j : x_j \dots x_{j+n-1} = B\}.$$

Put

$$L(B) = \{\ell_1(B), \ell_2(B), \ell_3(B), \dots\}$$

with increasing order, i.e., $\ell_k(B) < \ell_{k+1}(B)$ for all $k \geq 1$. We obtain the sample mean of $E[\log R_n \,|\, x_1^n = B]$ by

$$\frac{1}{M} \sum_{i=1}^{M} \log\left(\ell_{i+1}(B) - \ell_i(B)\right),$$

where M is the sample size.

For each 2^n blocks we compare the theoretical values and the sample mean values where the sample size for every generator is 100,000 in our experiments and $n = 14$. The standard Z-test for $\log R_n$ is applied for these all 2^n blocks. We obtain the sample averages of $\log R_n$ by observing the recurrence times of each block and we compare them with $E[\log R_n]$ and $Var[\log R_n]$.

Table 2 show the number of blocks such that $Z < -2.57$, $Z < -1.96$, $Z > 1.96$, and $Z > 2.57$. Note that the total number of block is $2^{14} = 16384$. If the absolute value of Z-value is larger than 1.96 and 2.57, then the corresponding generator fails the test for the corresponding n with statistical confidence of 95% and 99%, respectively.

TABLE 3. The variance test for Type I blocks for Z-values ($n = 14$)

Generator	Type I-1	Type I-2	Type I-3	Type I-4	Type I-5	Type I-6
Fishman	0.66	0.79	0.97	0.76	0.67	0.68
ICG	1.22	0.97	0.91	0.95	1.16	0.87
Ran0	0.87	0.72	0.72	0.64	0.71	0.86
Ran1	0.78	0.93	0.79	0.78	0.88	0.76
Ran2	1.30	0.90	1.13	0.96	1.17	0.84
Ran3	0.96	1.07	0.93	0.97	1.05	0.96
F90	0.92	1.14	0.77	0.99	1.02	0.90

TABLE 4. The variance test for Type II blocks for Z-values ($n = 14$)

Generator	type II-1	Type II-2	Type II-3	Type II-4	Type II-5
Fishman	0.87	0.75	0.73	0.88	0.76
ICG	1.04	1.06	1.00	0.95	0.84
Ran0	0.95	0.75	0.73	0.93	0.79
Ran1	0.93	0.83	0.91	0.95	0.78
Ran2	1.16	0.89	0.88	1.26	1.27
Ran3	0.91	0.90	1.02	0.96	1.16
F90	1.03	1.03	1.00	1.01	1.13

In Table 2, Mean and Variance denote the mean and variance of Z-values among the all 2^{14} blocks. For ideal generators have the mean and variance near 0 and 1 respectively. Apparently, Randu, ANSI, and MS seem to fail the test.

Since the Z-values among 2^n blocks are highly correlated, we need to reduce the correlation among the sample values. For example, the return time of 00000000000000 and 00000000000001 are highly related since there is a big chance that 00000000000001 appears right after 00000000000000. Therefore, we test the variance test on only special kind of blocks. A binary block will be regarded as an integer in binary expansion, i.e., for $B =$ 00000001000010 we will say $B = (1000010)_{(2)} = 130$. We will take a set of blocks of the form $B \equiv a \pmod b$ or $B = bk + a$ for some integer $k \geq 0$, of which blocks are not easily overlaped with each other. If two blocks $B = bk + a$ and $B' = bk' + a$ are overlaped, then we have

$$B = CB_0, \qquad B' = B_0C'$$

for some ℓ-blocks C and C'. In this case, $2^\ell(bk + a) + m' = 2^{14}m + bk' + a$ for some m and m' with $0 \leq m, m' \leq 2^\ell - 1$. Note that m and m' are representation of C and C'. Therefore we have

$$(2^\ell - 1)a + m \equiv dm' \pmod b,$$

where $2^{14} \equiv d \pmod b$. Choose b as $2^{14} \equiv 1 \pmod b$. Then if $B = CB_0$ and $B' = B_0C'$ for some ℓ-blocks C and C', then we have

$$(2^\ell - 1)a \equiv m \pmod b, \quad \text{for some } 0 \le m \le 2^\ell - 1. \tag{3}$$

Then possible value of b for $2^{14} \equiv 1 \pmod b$ with $1 < b < 2^{14} - 1$ is 3, 43, 127, 129, 381 and 5461.

Choose $b = 127$. Then for each $a = 64, 72, 84, 106, 118, 126$ there is no ℓ with $1 \le \ell \le 6$ which satisfies (3). When $b = 129$, if we pick $a = 65, 83, 108, 120, 128$, then (3) is not satisfied for any ℓ with $1 \le \ell \le 6$. We say the block B is of type I-1 (respectively I-2, I-3, I-4, I-5 and I-6), if the integer obtained by the binary block is $64 \pmod{127}$ (respectively $72 \pmod{127}$, $84 \pmod{127}$, $106 \pmod{127}$, $118 \pmod{127}$ and $126 \pmod{127}$). Similarly B is of type II-1 (respectively II-2, II-3, II-4 and II-5), if the integer obtained by the binary block is $65 \pmod{129}$ (respectively $83 \pmod{129}$, $108 \pmod{129}$, $120 \pmod{129}$ and $128 \pmod{129}$).

The variance test of the Z-values over the blocks of each type is applied. The test results are presented in Table 3 and 4. Test I-1,2,3,4,5,6 and II-1,2,3,4,5 denote the tests on the blocks of type I-1,2,3,4,5,6 and II-1,2,3,4,5 respectively. The number of each type I-1, I-2, I-3, I-4, I-5, I-6 blocks is 129 and degree of freedom of the corresponding chi-distribution is $n - 1$ = 128. For each type I blocks, if the sample variance is bigger than 1.26 or less than 0.77, than it fails the variance test with 5% significance level and if the sample variance is bigger than 1.35 or less than 0.71, than it fails the variance test with 1% significance level. The number of each type II-1, II-2, II-3, II-4, II-5 blocks is 127 and the degree of freedom of the chi-distribution is $n - 1$ = 126. For each type II blocks, if the sample variance is bigger than 1.26 or less than 0.77, than it fails the variance test with 5% significance level and if the sample variance is bigger than 1.35 or less than 0.71, than it fails the variance test with 1% significance level. Tests for Randu, ANSI, and MS are skipped because they fail the previous test. In this test all generator which is made up of one linear congruential generator fail the test.

4. GENERATORS

The following is a list of PRNGs tested in Section 3. We generate binary sequences using the algorithms listed in Table 5. A linear congruential generator $LCG(M, a, b)$ means the algorithm given by

$$X_{n+1} \equiv aX_n + b \pmod M.$$

Randu is an outdated generator developed by IBM in the sixties. ANSI and Microsoft are the generators used in C libraries by American National Standard Institute and Microsoft, respectively. For a prime p, the inversive congruential generator $ICG(p, a, b)$ is that

$$X_{n+1} \equiv a\overline{X_n} + b \pmod p,$$

where $\overline{X}$ is the multiplicative inverse of x modulo p. The generators Ran0, Ran1, Ran2 and Ran3 are from [13]. Ran0 is the linear congruential generator by Park and Miller[12]. Ran1 is

TABLE 5. The tested random number generators

Name	Generator	Period
Randu	$LCG(2^{31}, 65539, 0)$	2^{29}
ANSI	$LCG(2^{31}, 1103515245, 12345)$	2^{31}
Microsoft	$LCG(2^{31}, 214013, 2531011)$	2^{31}
Fishman	$LCG(2^{31}-1, 950706376, 0)$	$2^{31}-2$
ICG	$ICG(2^{31}-1, 1, 1)$	$2^{31}-1$
Ran0	$LCG(2^{31}-1, 16807, 0)$	$2^{31}-2$
Ran1	Ran0 with shuffle	$> 2^{31}-2$
Ran2	L'Ecuyer's algorithm with shuffle	$> 2.3 \times 10^{18}$
Ran3	$X_n \equiv X_{n-55} - X_{n-24} \pmod{2^{31}}$	$\geq 2^{55}-1$
F90	Ran0 combined with shift register	$\sim 3.1 \times 10^{18}$

Ran0 with Bays-Durham shuffle. Ran2 is L'Ecuyer's generator[9] made up of

$$LCG(2147483563, 40014, 0) \text{ and } LCG(2147483399, 40692, 0)$$

with Bays-Durham shuffle. Ran3 is a subtractive lagged Fibonacci sequences. F90[14] is Ran0 combined with a Marsaglia shift register generator, which is the form of $X_{n+1} = X_n(I \oplus L^{13})(I \oplus R^{17})(I \oplus L^5)$, where $\oplus$ denotes the binary exclusive-or operation and L (resp. R) is the bitwise left-shift (resp. right-shift).

REFERENCES

[1] G.H. Choe, C. Kim, and D.H. Kim, *Applications of ergodic theory to pseudorandom numbers*, Bull. Korean Math. Soc. **35** (1998), 173–187.

[2] G.H. Choe and D.H. Kim, *Average convergence rate of the first return time*, Colloq. Math. **84/85** (2000), 159–171.

[3] G.H. Choe and D.H. Kim, *The first return time test of pseudorandom numbers*, J. Comput. Appl. Math. **143** (2002), no. 2, 263–274.

[4] J. Oliver, *A central limit theorem for non-overlapping return times*, J. Appl. Probab., **43** (2006), 32–47.

[5] M. Kac, *On the notion of recurrence in discrete stochastic processes*, Bull. Amer. Math. Soc. **53** (1947), 1002–1010.

[6] C. Kim, G.H. Choe and D.H. Kim, *Tests of randomness by the gambler's ruin algorithm*, Appl. Math. Comput. **199** (2008), 195–210.

[7] D.H. Kim, *The recurrence of blocks for Bernoulli processes*, Osaka J. Math. **40** (2003), no. 1, 171–186.

[8] I. Kontoyiannis, *Asymptotic recurrence and waiting times for stationary processes*, J. Theor. Probab. **11** (1998), 795–811.

[9] P. L'Ecuyer, *Efficient and portable combined random number generators*, Comm. ACM. **31** (1988), no. 6, 742–749.

[10] U. Maurer, *A universal statistical test for random bit generators*, J. Cryptology **5** (1992), 89–105.

[11] D. Ornstein and B. Weiss, *Entropy and data compression schemes*, IEEE Trans. Inform. Theory **39** (1993), no. 1, 78–83.

[12] S. Park and K. Miller, *Random number generators: good ones are hard to find*, Comm. ACM. **31** (1988), no. 10, 1192–1201.

[13] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd. ed., Cambridge Univ. Press, Cambridge, 1992.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in Fortran 90*, Cambridge Univ. Press, Cambridge, 1996.

[15] P. Shields, *Waiting times: positive and negative results on the wyner-ziv problem*, J. Theor. Probab., **6** (1993), 499–519.

[16] A.D. Wyner and J. Ziv, *Some asymptotic properties of the entropy of stationary ergodic data source with applications to data compression*, IEEE Trans. Inform. Theory **35** (1989), no. 6, 1250–1258.

[17] A.J. Wyner, *Strong Matching Theorems and Applications to Data Compression and Statistics*, Ph.D. thesis, Stanford University, Department of Statistics, 1993.

[18] A.J. Wyner, *More on recurrence and waiting times*, Ann. Appl. Probab. **9** (1999), no. 3, 780–796.

[19] J. Ziv and A. Lempel, *A universal algorithm for sequential data compression*, IEEE Trans. Inform. Theory **23** (1977), no. 3, 337–343.