Available online: September 15, 2017

Commun. Fac. Sci. Univ. Ank. Ser. A1 Math. Stat.
Volume 67, Number 2, Pages 64–81 (2018)
DOI: 10.1501/Commua1_0000000862
ISSN 1303–5991
http://communications.science.ankara.edu.tr/index.php?series=A1

MODIFICATIONS OF KNUTH RANDOMNESS TESTS FOR INTEGER AND BINARY SEQUENCES

ONUR KOÇAK, FATIH SULAK, ALI DOĞANAKSOY, AND MUHIDDIN UĞUZ

Abstract. Generating random numbers and random sequences that are indistinguishable from truly random sequences is an important task for cryptography. To measure the randomness, statistical randomness tests are applied to the generated numbers and sequences. Knuth test suite is the one of the first statistical randomness suites. This suite, however, is mostly for real number sequences and the parameters of the tests are not given explicitly.

In this work, we review the tests in Knuth Test Suite. We give test parameters in order for the tests to be applicable to integer and binary sequences and make suggestions on the choice of these parameters. We clarify how the probabilities used in the tests are calculated according to the parameters and provide formulas to calculate the probabilities. Also, some tests, like Permutation Test and Max-of-t-test, are modified so that the test can be used to test integer sequences. Finally, we apply the suite on some widely used cryptographic random number sources and present the results.

1. Introduction

Random numbers have an important role in various areas. From daily life cryptographic applications like cell phone, SSL [1] to military communication random numbers are vital. The quality of the random number generator is vital for the security level of the application. For example, if the key used in an encryption algorithm is not random, that is some bits of the key can be guessed with a probability higher than $\frac{1}{2}$, then the complexity for obtaining the ciphertext will be easier than the claimed security of the algorithm. Therefore, generating random numbers and random sequences that are indistinguishable from a truly random sequence is an important task. Random numbers are generated either from a deterministic or an non-deterministic generator. The term *random number generator(RNG)* generally refers to the non-deterministic random number generators. There are various true random number generators actively sold in the market [2, 3]. The

Received by the editors: August. 06, 2016; Accepted: June 12, 2017.
2010 *Mathematics Subject Classification.* 65C10.
Key words and phrases. Knuth test suite, Statistical randomness testing .

deterministic random number generators are called *pseudo-random number generators (PRNGs)* [4, 5]. For some reasons like regenerating the random number or the efficiency of the generator, the PRNGs are preferred over RNGs. Among with the advantages PRNGs are weaker than RNGs in terms of randomness of the output as they are deterministic. Therefore, the PRNGs should be tested to measure how their outputs are close to the the outputs of the RNGs. For this purpose, PRNGs are subject to statistical randomness tests.

A statistical randomness test compares a specific property of the sequence to that of a truly random sequence and produces an output value which indicates the randomness of the sequence. For example, in a random bit sequence, the number of ones and the number of zeros should be equal or close to each other. Frequency test [6] checks if the number of occurrences of ones and zeros within the sequence are as expected from a truly random sequence.

A single test is not enough to conclude randomness of a PRNG. The generator should be tested by various statistical randomness tests, each of which inspects a different aspect of a random sequence. Therefore, various tests are gathered together to form a test suite and applied to sequences. Knuth [7], NIST [6], Diehard [8], Dieharder [9], TestU01 [10] are examples of tests suites in the literature.

Knuth is one of the first researchers who published a test suite consisting of 11 tests in his book [7]. In this suite, the underlying theory of tests for real number sequences are given. Some of these tests are intended to be applicable to integer sequences as well. However, assumptions for real number sequences are not suitable for integer sequences and causes problems when testing these sequences. For example, Permutation Test assumes any successive terms cannot be equal and all the test probabilities are given under this assumption but the equality occurs with a non-negligible probability for integer sequences. In order to the make the suite suitable for integer and binary sequences, new combinatorial calculations should be made. Moreover, even if one tests a real number sequence, the test parameters like sequence length, alphabet size, block size and the like, are not given for most of the tests in the suite. Therefore, besides new calculations, corresponding test parameters should be given for each test for the suite to be applicable.

In this paper, we calculate the test probabilities for binary and integer sequences by considering the abovementioned problems. Moreover, we calculate χ^2 probabilities for all tests to have a similar evaluation approach with Knuth. We also give test parameters, necessary sequence lengths and corresponding probability values, regarding efficiency and applicability. As a result, we modify 9 tests of Knuth Test Suite so that the modified tests are applicable to binary sequences.

The paper is organized as follows. In Section 2 the notation used in the paper and preliminary information about the primitives used in the calculations are given. Then, in Section 3, the details of the tests are given. In Section 4 the application results are presented. Finally, Section 5 concludes the paper.

2. Preliminaries

In Knuth Test Suite, integer valued sequences are considered. However, in order to use Knuth Test suite for cryptographic purposes we consider binary sequences in the following manner. Assume that a binary sequence, S of length l, and a block size b are given. Then, partition the sequence into non-overlapping blocks of size b, and discard the remaining terms, if any. Each block is considered as base 2 representation of an integer in $\{0, 1, \ldots, 2^b - 1\}$. In this way, we obtain an integer sequence of length $l_b = \left\lfloor \frac{l}{2^b} \right\rfloor$ where the elements are from an alphabet of size $d = 2^b$. In other words,

$$S = s_1 s_2 \ldots s_l, \;\; s_i \in \mathcal{A}, \;\text{ for } 1 \leq i \leq l, \;\text{ and } \mathcal{A} = \{0, 1, \ldots, d-1\}.$$

For instance if the binary sequence

$$S = 10010100100111101$$

is given and the alphabet size for the test is 8 (or block size b is 3), then the sequence should be converted to 3-bit integer sequence:

$$S' = (100)_2(101)_2(001)_2(001)_2(111)_2 01 = 4, 5, 1, 1, 7.$$

Note that the partitioning is non-overlapping for all the tests mentioned in this paper. It is also trivial to convert any integer sequence to the d-bit integer sequence.

Some tests partition the sequence into blocks of t consecutive elements and consider the distribution of the blocks. In this case, n denotes the number of blocks.

$$\begin{aligned} S &= (s_1 s_2 \ldots s_t)(s_{t+1} \ldots s_{2t}) \ldots (s_{(n-1)t+1} \ldots s_{nt}) \\ &= b_1 b_2 \ldots b_n \end{aligned}$$

Moreover, some tests need to apply operations on the sequence multiple times.

Knuth evaluates the sequences using χ^2 goodness-of-fit test which compares the observations to the expected values using k bins [7]. The observed number of elements in each bin is compared to the expected number of elements. In order to apply χ^2 properly, each bin should have at least 5 elements. The test outputs a p-value which is the probability of getting the observed results given that the sequence is random. To decide if a sequence passes a test or fails, a limit called significance level, α, is specified. If the p-value is greater than or equal to α, the sequence is said to pass the test. In statistical randomness testing, generally, α is chosen to be 0.01 or 0.05.

In the probability calculations of some tests, the Stirling numbers of the second kind is used. Stirling numbers of the second kind is the number of ways to partition a set of g elements into h non-empty subsets and denoted by $\left\{ \begin{matrix} \mathrm{g} \\ \mathrm{h} \end{matrix} \right\}$. The Stirling number of the second kind $\left\{ \begin{matrix} \mathrm{g} \\ \mathrm{h} \end{matrix} \right\}$ can be computed as

$$\left\{ \begin{matrix} \mathrm{g} \\ \mathrm{h} \end{matrix} \right\} = \frac{1}{h!} \sum_{j=0}^{h} (-1)^{h-j} \binom{h}{j} j^n.$$

3. Knuth's Statistical Randomness Tests

In this chapter, the tests in the Knuth test suite is investigated in details. For some tests, major changes are proposed without changing the approach followed by Knuth. Moreover, we propose test parameters that are not given in [7] for all the tests mentioned in this work.

We cover all the tests in Knuth test suite except the Run Test and the Serial Correlation Test. In the Run Test, it is assumed that the successive elements cannot be equal. For real number sequences this assumption is reasonable, however, for integer sequences the successive elements can be equal with a non-negligible probability. Without this assumption, the required computations are quite difficult and the modification of run test, unlike other tests, is beyond the scope of this paper. Yet, there is an ongoing work to modify the run test for integer and binary sequences. The Serial Correlation Test, on the other hand, does not output a p-value and the output of this test is not comparable to the outputs of the other tests.

3.1. **Equidistribution (Frequency) Test.** Equidistribution test checks if number of occurrences of each element $a \in \mathcal{A}$ are as expected from a random sequence. Knuth proposed two methods to apply this test;

(1) Use the Kolmogorov-Smirnov test with $F(x) = x$ for $0 \leq x < d$.
(2) For each element a, $0 \leq a < d$, count the number of times a appeared in the sequence and then apply the χ^2 test with degree of freedom $k = d - 1$, where the expected probability of each bin is $p_a = \frac{1}{d}$.

In this work, we proceed considering the second method. In [7], no parameters are given for the alphabet size and the length of the sequence. In order to apply the χ^2 test properly, the size of the alphabet should be chosen accordingly with the length of the sequence. For example, if S is 128 bits, then d, the size of the alphabet, should be at most 4. Otherwise, the expected number of elements in each bin cannot exceed 5 and χ^2 test cannot be applied. In fact, for each bin to have at least 5 elements, we should have $l \cdot \frac{1}{d} \geq 5$, ie. $l \geq 5d$. Since each element is of size $log_2 d$ bit, the length of the sequence should be at least $5d \ log_2 d$ bits. Leaving a safe distance, Table 1 can be used to decide on the alphabet size d for a given sequence size.

The following is an example on how to apply the test and calculate the p-value. Let $S = 10001010110111110100100110110010$, with $l_b = 32$. According to Table 1, the alphabet size should be 4 ie. each element is 2-bit. Then, the counters for 2-bit

l_b	$l_b \leq 20$	$20 < l_b \leq 80$	$80 < l_b \leq 240$	$240 < l_b \leq 640$	$640 < l_b \leq 1600$
d	2	4	8	16	32

Table 1. Sequence Bit Length-Alphabet Size Table for Equidistribution Test

elements are $\#00 : 3, \#01 : 3, \#10 : 6, \#11 : 4$. Alternatively, one can convert the sequence into a 2-bit integer sequence $S' = 2, 0, 2, 2, 3, 1, 3, 3, 1, 0, 2, 1, 2, 3, 0, 2$ and count the number of occurrences of each element. The test value can be computed as

$$\begin{aligned} \chi^2 &= \sum_{i=1}^{k} \frac{(Observed_i - Expected_i)^2}{Expected_i} \\ &= \sum_{i=1}^{4} \frac{(Observed_i - 4)^2}{4} \\ &= 0.25 + 0.25 + 1 + 0 \\ &= 1.5 \end{aligned}$$

The p-value for degree of freedom $k = 3$ and the test value 1.5 is 0.6822. Assuming the significance level of $\alpha = 0.01$, the sequence passes the Equidistribution Test.

3.2. **Serial Test.** In Knuth test Suite, Serial test is an Equidistribution Test for pairs and hence it is equivalent of Equidistribution Test with alphabet size d^2. It checks whether the pairs of elements are equally distributed within the tested sequence or not. The test is proposed as follows: partition the sequence into non-overlapping subsequences of size two: $S_2 = (s_1, s_2), (s_3, s_4) \ldots (s_{2n-1}, s_{2n})$. Then, for each possible pair (q, r) with $0 \leq q, r < d$, count the number of occurrences of the pair (q, r) and apply χ^2 goodness-of-fit test with $d^2 - 1$ degrees of freedom and $\frac{1}{d^2}$ expected probability for each bin. Since there are $\frac{l}{2}$ pairs and each bin has the same probability, for χ^2 to be applicable, the inequality $\frac{l}{2}\frac{1}{d^2} \geq 5$ should be satisfied, which gives $l \geq 10d^2$. Therefore, the length of the sequence should be at least $10d^2\ log_2 d$ bits.

The suggested parameters for the Serial Test are given in Table 2.

l_b	$l_b \leq 80$	$80 < l_b \leq 480$	$480 < l_b \leq 2880$	$2880 < l_b \leq 15360$
d	2	4	8	16

Table 2. Sequence Bit Length-Alphabet Size Table for Serial Test

This test can be extended to triples or quadruples easily, however, l should be large enough or d should be taken small in order to get reasonable number of triples/quadruples.

3.3. **Gap Test.** This test examines the distribution of the lengths of the gaps among the elements of a specified set within the sequence. To apply the test, first, a subset $\mathbb{U}$ of $\mathcal{A}$ is fixed. Then, the number of gaps between the elements of $\mathbb{U}$ in the sequence S are counted according to their lengths. For example, assume $\mathcal{A} = \{0, 1, \ldots, 7\}$, $S = 7, 2, 4, 6, 2, 5, 2, 7, 4, 5, 6, 0, 7, 4, 1, 1, 7, 0, 4, 1$ and let $\mathbb{U} = \{a \mid a < 4, a \in \mathcal{A}\}$. If we mark the elements of $\mathbb{U}$ we get $S = 7, \mathbf{2}, 4, 6, \mathbf{2}, 5, \mathbf{2}, 7, 4, 5, 6, \mathbf{0}, 7, 4, \mathbf{1}, \mathbf{1}, 7, \mathbf{0}, 4, \mathbf{1}, 6$. The gaps between the elements of $\mathbb{U}$ are of length 2, 1, 4, 2, 0, 1, 1 in order. The number of gaps of size zero is 1, size one is 3, size two is 2 and size four is 1. Finally, the observed distribution of the length of the gaps are compared to the expected distribution applying χ^2 goodness-of -fit test and a p-value is obtained.

The following algorithm gives the expected probabilities of the length of the gaps.

Theorem 1. *Let $\mathcal{A}$ be an alphabet of size d and $\mathbb{U}$ be any nonempty subset of $\mathcal{A}$. Let S be a random sequence of elements of $\mathcal{A}$ and let $s_i \in \mathbb{U}$ for some i. Then, the probability that $s_{i+k} \notin \mathbb{U}$ for $k = 1, 2, .., r$ is*

$$p_r = \left(1 - \frac{|\mathbb{U}|}{d}\right)^r \frac{|\mathbb{U}|}{d}.$$

Proof. In order for a gap of length r to occur, after an element of $\mathbb{U}$, r elements from the set $\mathcal{A} \backslash \mathbb{U}$ should follow and to terminate the gap there must follow an element from $\mathbb{U}$:

$$u \underbrace{v \ldots v}_{\text{r}} \; u, \;\; u \in \mathbb{U}, v \in \mathcal{A} \backslash \mathbb{U}$$

Since an element from $\mathbb{U}$ will appear in the sequence with probability $\frac{|\mathbb{U}|}{d}$ the probability of the length of the gap to be r is $(1 - p_u)^r p_u$. □

For the example above, $p_u = \frac{4}{8}$. Therefore, the probabilities of the length of the gaps being 0, 1, 2 and 3 are $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ and $\frac{1}{16}$ respectively. So, since the number of total gaps is 7, the expected number of gaps are $\frac{7}{2}, \frac{7}{4}, \frac{7}{8}$ and $\frac{17}{16}$ for 0, 1, 2 and 3. Applying the χ^2 test with the expected and observed values we get the p-value as 0.183255.

For short sequences as above the probability of long gaps will be very small. On the other hand, for long sequences the number of lengths will be too many to handle. Therefore, it is a good idea to limit the number of lengths as $r = 0, 1, \ldots j - 1$ and $r \geq j$ for a proper j. The probability of the length of a gap to be greater than or equal to j is $(1 - p_u)^j$ as after the first j elements from the set $\mathcal{A} \backslash \mathbb{U}$, no matter next element belongs to $\mathbb{U}$ or not the size of the gap will be greater than or equal to j.

One should choose j, $\mathbb{U}$ and l so that, p_j and p_r, for $r = 0, 1, \ldots, j-1$, enables the application of χ^2 test. That is, the number of gaps of length r, for $r = 0, 1, \ldots, j-1$ and $r \geq j$ should be at least 5.

For example, considering $d = 256$, if one chooses $|\mathbb{U}| = 4$, then the probability of a gap of length 0 becomes $\frac{4}{256} = 0.015625$. In order to expect at least 5 gaps of length 0, the total number of total gaps should be at least $5 \cdot \frac{1}{0.015625} = 320$. g gaps require $g + 1$ elements from $\mathbb{U}$, therefore, for 320 gaps one needs 321 elements from $\mathbb{U}$. Since $|\mathbb{U}| = 4$, on average 4 elements from $\mathbb{U}$ will occur in 256 elements in the sequence. Therefore, for 320 gaps one needs a sequence of 19968 elements that is 159744 bits. Since the probabilities for longer gaps will be smaller, the required sequence length will be longer.

However, considering $|\mathbb{U}| = 16$ with $d = 256$ one gets more applicable results. In this case

$$\begin{aligned}
p &= \frac{16}{256} = \frac{1}{16} \\
p_0 &= \frac{1}{16} = 0.062500 \\
p_1 &= \frac{1}{16}\frac{240}{256} = 0.058593 \\
p_2 &= \frac{1}{16}\left(\frac{240}{256}\right)^2 = 0.054931 \\
p_3 &= \frac{1}{16}\left(\frac{240}{256}\right)^3 = 0.051498 \\
p_4 &= \frac{1}{16}\left(\frac{240}{256}\right)^4 = 0.048279 \\
p_{>4} &= \left(\frac{240}{256}\right)^5 = 0.724196.
\end{aligned}$$

Since the lowest probability is p_4, about $\lceil\frac{5}{0.048279}\rceil = 104$ gaps needed for χ^2 to be applicable. This makes 1680 elements and a 13440 bit sequence will be long enough which is more feasible than $|\mathbb{U}| = 4$ case. So, one can use the gap test with $d = 256$, $l > 13440$ bits, $|\mathbb{U}| = 16$, for instance $\mathbb{U} = \{x|x < 16\}$, and given probabilities above.

For shorter sequences, one may take $|\mathbb{U}|$ larger and consider less χ^2 bins. For instance, for a sequence of 1200 bits, take $|\mathbb{U}| = 64$, and consider the bins for $r = 0, 1, 2, 3$ and $r > 3$.

$$\begin{aligned}
p &= \frac{64}{256} = \frac{1}{4} \\
p_0 &= \frac{1}{4} = 0.25 \\
p_1 &= \frac{1}{4}\frac{192}{256} = 0.187500
\end{aligned}$$

$$\begin{aligned} p_2 &= \frac{1}{4}\Big(\frac{192}{256}\Big)^2 = 0.140625 \\ p_3 &= \frac{1}{4}\Big(\frac{192}{256}\Big)^3 = 0.105468 \\ p_{>3} &= \Big(\frac{240}{256}\Big)^4 = 0.316406. \end{aligned}$$

3.4. **Poker Test.** This test checks if the distribution of the number of distinct elements in a t-tuple is as expected from a random sequence. In [7], Knuth considers n groups of non-overlapping t successive elements and counts the number of t-tuples containing exactly r distinct elements where $r = 1, 2, \ldots, t$. The probability of a t-tuple to have exactly r distinct elements is as follows.

Theorem 2. *Let $\mathcal{A}$ be an alphabet of size d and $a_1a_2 \ldots a_t$ be a randomly chosen t-tuple from $\mathcal{A}^t$. Let $\mathbb{U} = \{a_1, \ldots, a_t\} \supseteq \mathcal{A}$. Then for each r, $1 < r \leq t$, the probability that $\mathbb{U}$ contains r distinct elements is*

$$Pr(|\mathbb{U}| = r) = \frac{d(d-1)\cdots(d-r+1)}{d^t} \left\{ {t \atop r} \right\}$$

where $\left\{ {a \atop b} \right\}$ is the Stirling number of the second kind.

Proof.

$$\begin{aligned} Pr(|\mathbb{U}| = r) &= \frac{\text{choosing r distinct elements out of d}}{\text{All possible } t\text{-tuples}} \left\{ \begin{matrix} \text{Number of ways to} \\ \text{partition t-tuple} \\ \text{into r subsets} \end{matrix} \right\} \\ &= \frac{d(d-1)\cdots(d-r+1)}{d^t} \left\{ {t \atop r} \right\} \end{aligned}$$

□

One should choose d and t carefully in order for the test to be applicable to variety of sizes. If we choose $d = 256$ as the above tests, unless selecting t very large which will result in need for a very long sequence, the probabilities for $r = 1, 2, \ldots, t-2$ will be very small. This will lead to small number of bins in χ^2 test and, also, will increase the necessary length of the sequence to have at least 5 elements in each bin. In that case, for the alphabet size a divisor or a multiple of 8 will be a good choose for implementation purposes since one byte corresponds to 8 bits. So, we choose 4-bit alphabet, ie. $d = 16$, with $t = 8$. Using these parameters, the probabilities p_r

can be calculated as

$$\begin{aligned}
p_1 &= 3.7 \times 10^{-9} \approx 0 \text{ since the number of blocks will be smaller than } 10^9\\
p_2 &= 0.000007\\
p_3 &= 0.000756\\
p_4 &= 0.017299\\
p_5 &= 0.128143\\
p_6 &= 0.357091\\
p_7 &= 0.375885\\
p_8 &= 0.120820.
\end{aligned}$$

The χ^2 test will be applied with 5 bins where the first bin is "less than 5 distinct elements" and other "r distinct elements" each composes a bin: second bin covers "5 distinct elements", third bin is composed of "6 distinct elements" and so on. Since the least probable case, "less than 5 distinct elements", has probability 0.018062, in order to apply χ^2 one needs $\lceil \frac{5}{0.018062} \rceil = 277$ blocks of 8 4-bit elements which means one needs at least 8864 bit sequence.

3.5. **Coupon Collector Test.** Coupon Collector test examines the sequence by the length of the subsequences that have a complete set of alphabet elements. Starting from the first sequence element, one traces the sequence until all the alphabet elements are covered and records the length of the subsequence. For example let $\mathcal{A} = \{0, 1, 2, 3\}$ and $S = 1, 0, 2, 1, 2, 0, 3, 3, \ldots$. Marking the first occurrences of alphabet elements, $S = \mathbf{1}, \mathbf{0}, 1, \mathbf{2}, 2, 0, \mathbf{3}, 3, \ldots$, it is seen that the length of the shortest subsequence containing all the alphabet elements is 7. Then, resuming from the following element, again, finds the length of the subsequence covering all the alphabet elements and so on. When all the sequence is traced, the length of the subsequences are compared to those of a random sequence.The expected probability for a subsequence of length c that covers all the elements in the alphabet is given below.

Theorem 3. *Let $\mathcal{A}$ be an alphabet of size d. The probability that all elements of $\mathcal{A}$ appears in a sequence $a_1 a_2 \ldots a_c$, but not in $a_1 a_2 \ldots a_{c_1}$ is*

$$p_c = \frac{d!}{d^c} \left\{ \begin{matrix} c-1 \\ d-1 \end{matrix} \right\},$$

and the probability that the subsequences is of length greater than or equal to c is

$$p_{\geq c} = 1 - \frac{d!}{d^{c-1}} \left\{ \begin{matrix} c-1 \\ d \end{matrix} \right\}.$$

Proof. Now notice that, since the last element completes the collection, it should not appear previously in the subsequence. That is, this element only occurs one and its the last position. Fixing the last element, we left with a subsequence of length

$c-1$, containing $d-1$ distinct elements. The number of distinct such sequences is equal to the number of onto functions from a set of size $c-1$ to a set of size $d-1$, which is $(d-1)! \left\{ {c-1 \atop d-1} \right\}$. Considering the last element is chosen from a set of size d, the number of distinct subsequences containing all d elements is $d(d-1)! \left\{ {c-1 \atop d-1} \right\}$. Since there are overall d^c subsequences, the probability of such a subsequence is

$$p_c = \frac{d!}{d^c} \left\{ {c-1 \atop d-1} \right\}.$$

The probability of a subsequence of length greater than or equal to c is the complement of the probability that a sequence of length $c-1$ containing all d elements in any order. This includes all subsequences containing d distinct elements from a subsequence of length d to a subsequence of length $c-1$. The probability of a subsequence of length $c-1$ containing d distinct elements is equal to the number of onto functions from a $c-1$-element set to a d-element set. So, the probability of such a subsequence is $p_{\tilde{c}} = \frac{d!}{d^{c-1}} \left\{ {c-1 \atop d} \right\}$. Therefore, the probability of a subsequence of length greater than or equal to c containing d distinct elements is $1 - p_{\tilde{c}} = 1 - \frac{d!}{d^{c-1}} \left\{ {c-1 \atop d} \right\}$. □

When considering the $d = 256$ again, computing the Stirling numbers becomes infeasible. Therefore, we need to decrease the alphabet size. Similar to the Poker Test case, the best candidate for d is 16. For the case $d = 16$, the bin values and the probabilities are given below where p_{i-j} is the probability that the length of the sequence covering all the alphabet elements is between i and j, inclusive.

$$\begin{aligned}
p_{16-34} &= 0.107625\\
p_{35-38} &= 0.085983\\
p_{39-42} &= 0.100841\\
p_{43-46} &= 0.104948\\
p_{47-50} &= 0.100590\\
p_{51-54} &= 0.090983\\
p_{55-59} &= 0.096727\\
p_{\geq 60} &= 0.312300
\end{aligned}$$

One can apply an 8-bin χ^2 goodness-of-fit test using the above probabilities. Since the lowest probability is $p_{35-38} = 0.085983$, the number of collections should be at least $\lceil \frac{5}{0.085983} \rceil = 59$. In the worst case, each subsequence containing a collection is at most 60 elements long, or one can stop searching for a collection after 60th element as the bin for 60 and any length longer then 60 are the same. Therefore, the sequence is 3540 elements long which is corresponding to 14160 bits.

3.6. **Permutation Test.** The Knuth Permutation Test focuses on the frequencies of the the arrangements of the elements within a block. Each block can be arranged in different ways considering the lexicographic ordering. For example, (4 3 0 1) and (9 7 4 5) have the same lexicographic ordering. Test compares the observed frequencies of the arrangements to the expected frequencies for a random sequence.

First, the sequence is divided into blocks of size t. In [7], Knuth assumes the sequence is a real number sequence and it is not expected to have a repetition within a block. It is assumed that each block can be arranged in one of $t!$ permutations. Counting the frequencies of each permutation, one can apply a χ^2 test with bin probability $\frac{1}{t!}$ for each bin. However, it is very likely that in an integer sequence there will be elements that will appear more than once within a block. In order to have an integer sequence that does not likely to contain repetitions within t-element blocks, the elements should be very large which makes the sequence too long. Another idea is to reduce the size of the blocks which in turn reduce the sensitivity of the test.s

Here, we propose another method to check the frequencies of the permutations without changing the notions in [7]. Again consider $d = 256$ and let t=4. The probability of occurring 4 distinct elements within a block is $\frac{256}{256}\frac{255}{256}\frac{254}{256}\frac{253}{256} = 0.976729$. Each 24 permutation of 4 distinct elements can occur with probability $p = \frac{0.976729989}{24} = 0.040697$ and repetition within a block occurs with probability $1 - 0.976729 = 0.023270$. So, applying the χ^2 test with 25 bins, 24 bins for non-repeating blocks and one for repeating blocks one can compare the sequence to a random sequence. To apply the χ^2 test one needs at least $\lceil \frac{5}{0.023270} \rceil = 215$ blocks of 4 elements, therefore, the length of the sequence must be at least $215 \cdot 4 \cdot log_2 256 = 6880$ bits.

3.7. **Max-of-t Test.** In [7], the Max-of-t Test is proposed to test the maximal elements within blocks of size t in order to check for randomness. The proposed test partitions the sequence into non-overlapping blocsk of t, and applies the Kol-mogorov-Smirnov test to the maximal elements of the sequences. However, Kolmogorov-Smirnov test is applied for examining a random sample from some unknown distribution to see the normality of the sample and it is less powerful than χ^2 goodness-of-fit test. Another option given in [7] is applying the Equidistribution Test to the maximal elements. Yet, the probabilities of maximal element to be 0 or $d-1$ are not equal. Therefore, one should consider each probability while applying the Equidistribution Test. Setting the parameters d and t, we find the probabilities of the maximum element to be exactly m within a block of t and to be smaller than or equal to m. This way one can apply χ^2 test with given probabilities and bin values.

Theorem 4. *Let $\mathcal{A}$ be an alphabet of size d.Then the the probability of maximum element to be less then or equal to m in a block of t terms is*

$$p_{(max \leq m)} = \frac{(m+1)^t}{d^t}.$$

Proof. Including "0", there are $m+1$ numbers less than or equal to m. In order for the maximum of t elements to be less than or equal to m, each of t elements can be one of $m+1$ numbers, ie. there are $(m+1)^t$ such blocks of t. Therefore, the probability of maximum element to be less then or equal to m is

$$p_{(max \leq m)} = \frac{(m+1)^t}{d^t}.$$

□

Moreover, the probability of maximum to be exactly m is

$$\begin{aligned} p_{(max=m)} &= p_{(max \leq m)} - p_{(max \leq m-1)} \\ &= \frac{(m+1)^t - m^t}{d^t} \end{aligned}$$

Again considering $d = 256$ and $t = 4$, one can use the bin values given in Table 3. For the χ^2-test to be applicable the least probable bin, last bin in this case, should have at least 5 elements. Therefore, there should be $\lceil \frac{5}{0.188169744} \rceil = 28$ blocks of 4 8-bit elements which sums up to 896 bits. So the sequence should be at least 896 bits to apply the Max-of-t test.

m	Bin Probability
$m \leq 170$	0.199078601
$171 \leq m \leq 203$	0.204158801
$204 \leq m \leq 225$	0.204161350
$226 \leq m \leq 242$	0.204431504
$243 \leq m$	0.188169744

Table 3. Bin boundaries and probabilities for Max-of-t Test

3.8. **Collision Test.** Collision test checks if the number of collisions in predefined parts of the sequences is as expected from a random sequences. In this test, the number of collisions are counted and the result is compared to the expected number of collisions.

The idea is similar to throwing balls into urns: if a ball lands in a nonempty urn, a collision is said to occur. If there are m urns and n balls then the probability of c collisions can be calculated as follows.

Theorem 5. *If n balls are thrown into m urns at random, the probability of occuring exactly c collisions is*

$$P\{C=c\} = \frac{m(m-1)\cdots(m-(n-c-1))}{m^n} \left\{ {n \atop n-c} \right\}. \tag{1}$$

Proof. In order for exactly c collisions to occur, first, n balls should land in $n-c$ distinct urns guaranteeing the number of collisions does not exceed c. There are $m(m-1)\cdots(m-(n-c-1))$ ways to choose $n-c$ urns out of m^n. Now each of $n-c$ urns have a single ball in it. Then, the remaining c balls can land in any of these urns, urns containing a single ball, in any order. For instance all the remaining c balls can land in the same urn or each ball may land in distinct urns. This is the partitioning of n balls into nonempty $n-c$ subsets, which is the Stirling number of the second kind, $\left\{ {n \atop n-c} \right\}$. Therefore, the probability of c collisions is

$$P\{C=c\} = \frac{m(m-1)\cdots(m-(n-c-1))}{m^n} \left\{ {n \atop n-c} \right\}.$$

□

For the randomness test, similarly, if the specified portions of two sequences are equal, a collision is said to occur and the probability in Equation 1 also applies to the test. In this case, the number of urns is the number of all possible subsequences in the predefined portion of the sequence. For example, consider the first 10 bits of the sequences. The number of "urns" is all possible 10 bit subsequences which is 2^{10}. The balls correspond to the distinct sequences to be tested.

Knuth suggests taking $m = 2^{20}$ and $n = 2^{14}$ which means taking 2^{20} sequences and counting the collisions in the predefined 14 bits of these sequences. For the sake of simplicity, one can take the first 14 bits or the last 14 bits of the sequence, but any set of fixed 14 bits of the sequence can be selected to inspect the collisions.

For the suggestions of Knuth, $m = 2^{20}$ and $n = 2^{14}$, the probabilities of collisions are given in Table 4. After counting the collisions in 2^{20} sequences, if the number of collisions is less than or equal to 101, the However, in this setting, one can just get a very inaccurate idea about the sequence by finding the interval in which the number of collisions lies. Therefore, applying the test on a series of sequences and getting a convenient result becomes inapplicable. In order to overcome this problem in a similar way with the previous tests, we calculate the collision probabilities and construct χ^2 bins. Using the bins one can apply χ^2 goodness-of-fit test and produce

# of Collisions	≤ 101	≤ 108	≤ 119	≤ 126	≤ 134	≤ 145	≤ 153
Probability	0.009	0.043	0.244	0.476	0.742	0.946	0.989

Table 4. Bin boundaries and probabilities for Collision Test

a p-value. The boundaries of the bins for $m = 2^{20}$ and $n = 2^{14}$ case are given in Table 5.

Moreover, taking 2^{20} distinct sequences is outside the scope of testing the randomness of a sequence. In fact, it is in the scope of testing a random number generator. Therefore, it is more convenient to partition the sequence into blocks instead of taking distinct sequences. For the given probabilities, in order to apply a proper χ^2 test, the number of experiments should be at least $\lceil \frac{5}{0.88373} \rceil = 56$. So, instead of taking a set of 2^{20} distinct sequences, one needs to partition the sequence into $56 \cdot 2^{20}$ blocks of 14 bits which suggests a sequence of 822083584 bits. In this case, one should divide the sequence into 56 subsequences, partition each subsequence into 2^{20} blocks and count the number of collisions in each subsequence. An alternative case for shorter sequences is taking $m = 2^{16}$ and $n = 2^{10}$. In this case, to apply the χ^2 test, the number of experiments should be at least $\lceil \frac{5}{0.141034} \rceil = 36$. Therefore, one needs $36 \cdot 2^{16}$ blocks of length 10 bits which makes 23592960 bits. Table 6 shows the boundaries and the probabilities for $m = 2^{16}$ and $n = 2^{10}$ case. Birthday Spacing Test

# of Collision	Probability
0-113	0.106253
114-118	0.109894
119-121	0.088373
122-124	0.100719
125-127	0.106608
128-130	0.104977
131-133	0.096322
124-137	0.106367
138-142	0.091574
143-16384	0.088913

Table 5. Collision Test χ^2 bin probabilities for $m = 2^{20}$ and $n = 2^{14}$

# of Collision	Probability
0-5	0.192924
6-7	0.259222
8	0.141034
9-10	0.223346
11-1024	0.177158

Table 6. Collision Test χ^2 bin probabilities for $m = 2^{16}$ and $n = 2^{10}$

The Birthday Spacing Test examines the randomness of the sequence by checking the number of equal differences between selected sequence elements. In this test, a number of sequence elements are selected, sorted, and the differences between each consecutive element are calculated. Then, the number of equal differences are compared to the expected number of equal differences. For example, let $S = 9, 5, 6, 1, 16, 24, 2, 13, 34, 29$ and consider the $4^{th}, 5^{th}, 9^{th}$ and 10^{th} elements: 1, 16, 34, 29. Sorting the elements we get S'=1, 16, 29, 34. The differences between the elements are $G = 16 - 1, 29 - 16, 34 - 29$ ie., $G = 15, 13, 15$. There are two equal differences, which means one collision occurs in differences. The test resembles the collision test and throwing balls into urns phenomenon with days of the year as urns and birthdays as balls. Since the elements of the alphabet are considered as the days of the year and the sequence elements are the birthdays, the name of the test is the birthday spacing test.

Knuth suggests to use $m = 2^{25}$ days for $n = 512$ birthdays. This setting, for bit sequences, is corresponding to taking 512 elements of 25 bits each, computing the differences between the consecutive elements. The probabilities for the number of colliding differences are given in Table 7. Using these probabilities one can apply a χ^2 test for goodness-of-fit.

# of Equal Spacings	0	1	2	3 or more
Probability	0.368801	0.369035	0.183471	0.078692

Table 7. The probabilities for Birthday Spacing Test

Similar to the Collision Test, in order to test the sequence, instead of taking distinct sequences, we take a sequence and partition the sequence according to the bit length of the “birthdays”. In order to apply the χ^2 test properly, one needs to make $\lceil \frac{5}{0.078692} \rceil = 64$ experiments each needs 2^{25} blocks of 9 bits long. Therefore, one needs $2^{25} \cdot 64 \cdot 9 \approx 2^{34}$ bits of data. In [7], advises to repeat the process 1000 times instead of 64 which increases the data size to 2^{40} assuming each sequence is 9 bits long.

4. Application

In this section we present the results of Knuth Test suite on various sequences. The primary aim of the section is to show the applicability of the suite on integer, and therefore on binary, sequences.

We applied the suite on $\pi, e, \sqrt{2}, log(2)$ and Riemann Zeta function $\zeta(3)$. For these numbers, we excluded the integer parts and test the sequence of 1.000.000 digits to the right of the decimal point. Moreover, we generate sequences, that have the same size with the previous sequences, by concatenating the SHA-256 [11] and MD-5 [12] hash values of successive integers starting from 0. Another sequence is generated by using the “random” utility of C#. Then, we generate a new sequence by giving a 1% “1” bias to this sequence. This way, test our parameters for frequency related tests. When testing the suite, we apply some tests twice with distinct parameters. The test parameters can be found in Table 8.

The results can be seen in Table 9. According to these results, all the non-biased sequences can be considered to be random. For the biased sequence, Frequency, Serial, Gap and Max-of-t tests output p-values less than 0.01 indicating the non-randomness as expected.

Test	Parameters
Frequency 1	$d = 256$
Frequency 2	$d = 2^{24}$
Serial	$d = 256$
Gap	$d = 256, \|\mathbb{U}\| = 16$
Poker	$d = 16, t = 4$
Coupon Coll	$d = 16$
Max-Of-t 1	$d = 256, t = 4$
Max-Of-t 2	$d = 2^{16}, t = 6$
Permutation 1	$d = 256, t = 4$
Permutation 2	$d = 2^{16}, t = 5$
Collision	$m = 2^{16}, n = 2^{10}$
Birthday Sp.	$m = 2^{25}, n = 512$

Table 8. Application Test Parameters

	PI	**E**	**Sqrt(2)**	**Log(2)**	**Golden Ratio**	**Zeta(3)**	**MD5**	**SHA256**	**C# Random**	**C# Biased**
Frequency 1	0,940520	0,174365	0,401369	0,551351	0,039588	0,046532	0,942153	0,509073	0,261939	0
Frequency 2	0,964781	0,261258	0,030199	0,931099	0,570073	0,506992	0,265583	0,221097	0,847772	0,001599
Serial	0,384719	0,052247	0,702899	0,980707	0,131627	0,024494	0,993911	0,231856	0,752165	0
Gap	0,709093	0,305874	0,440585	0,754035	0,348360	0,661038	0,723083	0,444013	0,979131	0
Poker	0,699648	0,956847	0,741170	0,560399	0,422498	0,957892	0,933983	0,355740	0,385174	0,002392
Coupon Coll	0,325971	0,213433	0,621810	0,853074	0,560253	0,837512	0,228078	0,519568	0,188181	0,139930
Max-Of-t 1	0,055390	0,267757	0,551455	0,599732	0,701230	0,150366	0,264187	0,576693	0,312611	0
Max-Of-t 2	0,101844	0,567233	0,665188	0,657665	0,765619	0,548888	0,351747	0,809745	0,020687	0
Permutation 1	0,118599	0,413592	0,388108	0,901025	0,953106	0,365188	0,347413	0,048359	0,559867	0,715372
Permutation 2	0,123178	0,639895	0,905754	0,937968	0,951257	0,069539	0,182614	0,591102	0,379035	0,214872
Collision	0,230030	0,640728	0,935599	0,769927	0,435727	0,698075	0,042924	0,044239	0,564757	0,343109
Birthday Sp.	0,042169	0,935038	0,249442	0,450414	0,060426	0,934736	0,135028	0,054025	0,764874	0,856611

Table 9. Test results of Knuth Test Suite for some mathematical constants and sequences

5. Conclusion

Knuth Test Suite [7] is one of the first statistical randomness test suites. The suite is well formed and the statistical basis of the test is well established. However, the suite is designed primarily to test real number sequences. The assumption given in the suite, that the tests could be applied to the integer sequences misses some points and some tests cannot be applied to integer sequences.

Moreover, the tester is assumed to have a knowledge over statistics and combinatorics that the test parameters and probability calculations are not given excluding one or two exceptions.

In this work, we review all the tests in Knuth Test Suite and excluding the Run Test and the Serial Correlation Test, we give test parameters in order for the tests to be applicable to integer sequences and make suggestions on the choice of these parameters. We clarify how the probabilities used in the tests are calculated according to the parameters and provide users to calculate the probabilities they need without any knowledge of statistics or combinatorics.

Also, some tests, like Permutation Test and Max-of-t-test, are reviewed so that the test can be used for integer sequences.

Finally, we apply the suite on some widely used cryptographic random number sources and present the results.

As a future work, the relations between Knuth Test Suite and NIST Test Suite will be investigated.

References

[1] P. K. A. Freier, P. Kocher, The secure sockets layer (ssl) protocol version 3.0 (2011). doi:10.17487/RFC6101.
URL <http://www.rfc-editor.org/info/rfc6101>

[2] Intel Corporation, Intel Digital Random Number Generator (DRNG): Software Implementation Guide, Revision 1.1 (2012).

[3] Comscire quantum number generators.
URL http://comscire.com/cart/

[4] M. Matsumoto, T. Nishimura, Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Trans. Model. Comput. Simul. 8 (1) (1998) 3–30. doi:10.1145/272991.272995.
URL http://doi.acm.org/10.1145/272991.272995

[5] L. Blum, M. Blum, M. Shub, A simple unpredictable pseudo random number generator, SIAM J. Comput. 15 (2) (1986) 364–383. doi:10.1137/0215025.
URL http://dx.doi.org/10.1137/0215025

[6] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications, Tech. rep., NIST (2001).
URL http://www.nist.gov

[7] D. E. Knuth, The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[8] G. Marsaglia, The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness (1996).
URL http://stat.fsu.edu/pub/diehard

[9] R. G. Brown, Dieharder: A random number test suite (2013).
URL http://www.phy.duke.edu/~rgb/General/dieharder.php

[10] P. L'Ecuyer, R. Simard, Testu01: A c library for empirical testing of random number generators, ACM Trans. Math. Softw. 33 (4) (2007) 22. doi:http://doi.acm.org/10.1145/1268776.1268777.

[11] Q. H. Dang, Fips 180-4, secure hash standard, Tech. rep., NIST (2012).

[12] R. Rivest, The md5 message-digest algorithm, in: RFC 1320, 1992.

Current address: Onur KOÇAK: TUBITAK BILGEM UEKAE, Turkey
E-mail address: `onur.kocak@tubitak.gov.tr`
ORCID: `http://orcid.org/0000-0001-5744-4727`
Current address: Fatih SULAK: Department of Mathematics, Atılım University, Ankara, Turkey
E-mail address: `fatih.sulak@atilim.edu.tr`
ORCID: `http://orcid.org/0000-0002-5220-3630`
Current address: Ali Doğanaksoy: Department of Mathematics, Middle East Technical University, Ankara, Turkey
E-mail address: `aldoks@metu.edu.tr`
ORCID: `http://orcid.org/0000-0002-3055-9863`
Current address: Muhiddin Uğuz: Department of Mathematics, Middle East Technical University, Ankara, Turkey
E-mail address: `muhid@metu.edu.tr`
ORCID: `http://orcid.org/0000-0003-2344-503X`