SECURE PASSWORD GENERATION THROUGH STATISTICAL RANDOMNESS TESTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYCAN USLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2017

Approval of the thesis:

SECURE PASSWORD GENERATION THROUGH STATISTICAL RANDOMNESS TESTS

submitted by **AYCAN USLU** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** _____

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Department of Mathematics, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Zülfükar Saygı
Department of Mathematics, TOBB University of Economics and Technology _____

Assoc. Prof. Dr. Ali Doğanaksoy
Department of Mathematics, METU _____

Assist. Prof. Dr. Fatih Sulak
Department of Mathematics, Atılım University _____

Assist. Prof. Dr. Elif Saygı
Mathematics and Science Education, Hacettepe University _____

Assist. Prof. Dr. Oğuz Yayla
Department of Mathematics, Hacettepe University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: AYCAN USLU

Signature :

ABSTRACT

SECURE PASSWORD GENERATION THROUGH STATISTICAL RANDOMNESS TESTS

Uslu, Aycan
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

September 2017, 44 pages

Both symmetric and asymmetric cryptographic algorithms must firstly be robust against brute force. The key needs to be choosen uniformly and randomly from the key space. It is possible to assure randomness by using statistical randomness tests which are also critical for other cryptographic issues as well. There is still an issue to be elaborated: the most well-known tool for attacking againts passwords namely dictionary attacks. These attacks are based on trying all keys from a particular subspace of the key space, which are composed of words from daily life and their variations. In this study we focus on the randomness of the keys but we are not interested with latter issue that is dictionary attacks. The one who use our tests to generate key must check it regarding specified dictionaries.

Keywords : Statistical Randomness Tests, Distribution Functions

ÖZ

İSTATİSTİKSEL RASTGELELİK TESTLERİ İLE GÜVENLİ PAROLA ÜRETİMİ

Uslu, Aycan
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy

Eylül 2017, 44 sayfa

Simetrik ve asimetrik şifre algoritmalarının öncelikle brute force'a dayanıklı olması beklenmektedir. Yani, belirli bir anahtar kümesi üzerinde yapılacak ataklara dayanıklı olması gerekmektedir. Bu da; seçilen anahtarın, uygun büyüklükteki bir anahtar uzayının herhangi bir yerinden eşit olasılıkla seçilmiş olması ile sağlanabilmektedir. Bunu garanti etmenin yolu başka kriptografik meselelerde de önem taşıyan istatistiksel rastgelelik testlerinin kullanılmasından geçer. Anahtarın, anahtar uzayından her anahtara eşit şans vererek seçilmiş olması brute force'a karşı dayanıklılığı sağlar. Bunun dışında anahtarlara(parolalara) atak yapmak için kullanılan en bilinen yöntemlerden bir diğeri sözlük saldırılarıdır. Bu ataklar, günlük hayattan seçilmiş kelimeler ve bunların varyasyonlarıyla oluşturulmuş anahtar uzayının alt uzaylarındaki bütün anahtarların denenmesi suretiyle gerçekleşir. Bu çalışmada anahtarların rastgelelik kriterlerine uygun olarak seçilmiş olması gözetilmiş, sözlük atakları gözardı edilmiştir. Bu nedenle, bu tezde anlatılan şekilde üretilen anahtarların belli sözlük testlerinden de geçirilmesi gerekmektedir.

Anahtar Kelimeler : İstatistiksel Testler, Dağılım Fonksiyonu

To My Mom

ACKNOWLEDGMENTS

This thesis is completed by the contributions of my advisor, Assoc. Prof. Dr. Ali Doğanaksoy. Firstly, I would like to express my gratefulness to him for his patience, long time support and supervision to me. I would also like to thank Assoc. Prof. Dr. Zülfükar Saygı, Assist. Prof. Dr. Fatih Sulak, Assist. Prof. Dr. Oğuz Yayla, Assist. Prof. Dr. Elif Saygı for revising this study during the jury process.

I would like to extend my sincerest thanks to my dear friends, Melek Mutioğlu Özkesen and Pınar Çomak for their friendship and unyielding support. They try to ease my thesis process in various ways. I would also thank to Burak Kaya and Ali Özkesen for their technical and psychological supports.

Special thanks are due to Ahmet Çetintaş for all the support and encouragement he gave me and his endless help in implementation of tests.

I owe biggest thanks to my family, my mom, Nursel Uslu, my dad, Ahmet Uslu, and my sisters, Nurcan Uslu Fazlı and Ayşen Uslu. I dedicated this study to them whose prays, good wishes and conditions have always worked for me that I am sure. This process would be very tough for me without their willing support. I also thank to God for my nephew and niece, Emre and Zeynep, who gave me a full of joy during this long thesis process.

This thesis has been supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) 2210 National Graduate Scholarship Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

TRNG	True Random Number Generator
PRNG	Pseudorandom Number Generator
σ	binary sequences
n	length of the binary sequence
$\tilde{\sigma}$	b-bit integer sequence corresponding to σ
b	integer bit size
l	length of the integer sequence, $l = n/b$

CHAPTER 1

Introduction

1.1 Random Sequences

Although it may look simple at first sight to give a definition of what a random number is, it proves to be quite difficult in practice.

A random number is generated by a unpredictable process, in which outcomes cannot be reliably reproduced later. In other sense, a kind of black box called a random number generator can accomplish this task, too. However, we cannot prove whether a singular random number was produced through a random number generator without examining sequences of numbers generated by the generator. Random sequences have specific properties as below:

- **Unpredictability:**Unpredictability: it means that knowing the first t element of a sequence does not inform anything about the next element of the sequence.
- **Uniformity:** In a sequence, 0's and 1's should be available in approximately equal numbers.
- **Independence:** Each term of the sequence is generated independently of the other terms.

It make us question about when a particular number or output string can be called unpredictable or uniformly distributed.

1.2 Random Number Generator

A random number generator is an algorithm, based on an initial seed or by means of continuous input, generates a sequence of numbers or bits. True Random Generators, abbreviated as TRNGs, output the results of a physical experiment which is considered to be random, like radioactive decay or the noise of a semiconductor diode. Outputs of

a deterministic algorithm which resembles a true number generator are called pseudo-random numbers. The specific feature of these generators is to use a numerical algorithm in order to generate a sequence of truly random numbers. In some cases, PRNGs use TRNGs with an additional algorithm which lead a sequence work like real random numbers.

TRNGs have some problems. Firstly, they are often biased, which means their output might include more ones than zeros, results that it does not correspond to a uniformly distributed random variable.In addition to this, some TRNGs are really expensive and requires an additional hardware device. They are usually too slow for applications we work. On the other hand, PRNGs do not need an additional hardware and they are faster than TNRGs. Moreover, their outputs accomplish the most necessary conditions of random numbers, like unbiasedness.

1.3 Statistical Randomness Test

Statistical tests are used to check whether the output sequences of a PRNG is statistically indistinguishable from the output of a truly random generator. They fulfill this task through calculating specific statistical quantities and comparing them with expected values. The expected values obtained from calculations are employed on the model of an ideal random number generator. Testing randomness is an empirical task for which there are various tests affirming any kind of imperfection in a sequence.

CHAPTER 2

Statistical Randomness Testing

Let Ω_n be the set of binary sequences of length n. Fixed n, an integer valued random variable defined on Ω_n that is X maps Ω_n into some subset of integers.

$$X : \Omega_n \to \mathbb{T}$$

Let X be a random variable on Ω_n and $\alpha \in (0, 1)$ is predetermined value, or one can call significance level. Respecting the random variable and its distribution function F does not enable us to call a sequence $\sigma \in X$ "good" or "bad". But we can call a sequence σ better than σ' if $Prob(X = X(\sigma)) \geq Prob(X = X(\sigma'))$. Then we can talk about, for example, the 'worst 20 sequence' or 'worst 20-percentage sequences' etc. We define the $'\alpha, X - worst'$ set to be the set of all sequences in Ω_n that we want to eliminate in test. We can call this set W_α and here is the way how we determine it:

Let probability density function of X be $F_n(k) = Prob(X = k)$ such that

$$F_n : \mathbb{T} \to [0, 1].$$

For any $u \in [0, 1]$ define $A_u \subset \Omega_n$ such that

$$A_u = \{\sigma \in \Omega_n | Prob(x = x(\sigma)) \leq u\}.$$

Let us define

$$Prob(A_u) = \sum_{\sigma \in A_u} Prob(x = x(\sigma)).$$

Given α, among all $u \in (0, 1)$ such that $Prob(A_u) \leq \alpha$ and call the largest one u_α. Then, $W_\alpha = A_{u_\alpha}$.

We decide whether a given sequence is $'\alpha, X - bad'$ or not using certain statistical properties of random variables. Vice versa, we can call sequence as $'1 - \alpha, X - good'$.

Here are some examples of distribution functions:

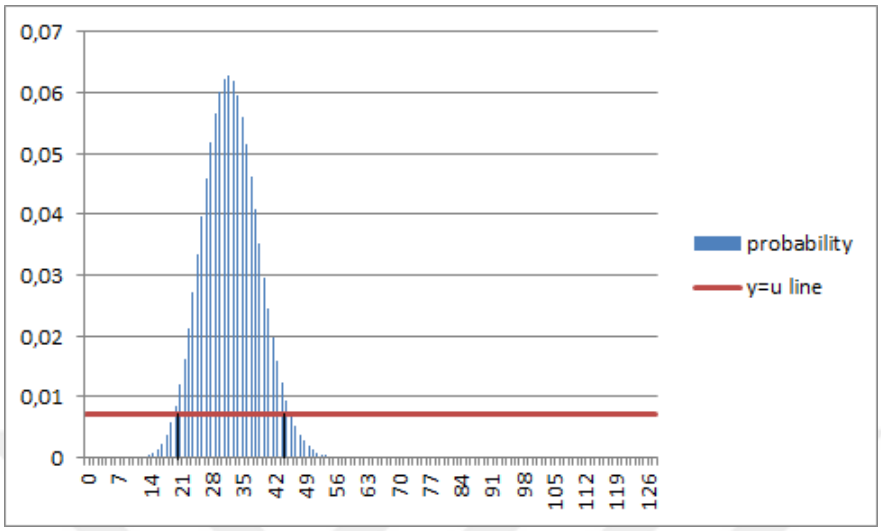

Figure 2.1: Example of Probability Distribution

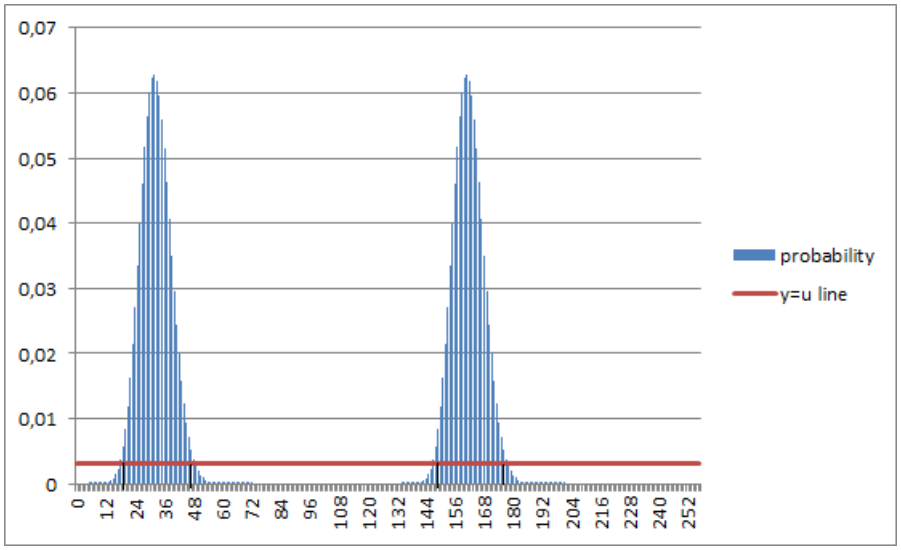

Figure 2.2: Example of Probability Distribution

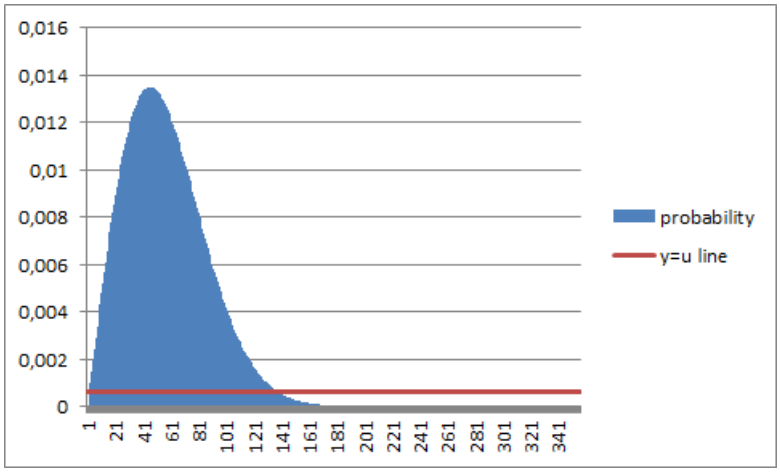

Figure 2.3: Example of Probability Distribution

In 2.3, all elements of W_α is in the one side of the distribution. So we call the test that we intend to apply as *one-sided test*. Likewise, *two-sided test* refer to the test for the distribution like in 2.1. In this study, we do not have any example of 2.2, but our method is still suitable for that kind of distribution.

2.1 Frequency (Weight) Test

Frequency Test is esteemed as the simplest test while we are testing randomness property. It measures the number of 1's in a sequence. There are $\binom{n}{w}$ n-bit sequences that has w weight, and the probability of that is;

$$Pr(W = w) = \frac{\binom{n}{w}}{2^n}.$$

Thus, as it has obviously been observed the the probability distribution function is the binomial distribution.

2.1.1 Recursion

Initial values are:

$$P_1(0) = 1, P_1(1) = \frac{1}{2},$$

for $n \geq 2$ and $k = 0$

$$P_n(0) = \frac{1}{2} P_{n-1}(0).$$

When $n \geq 2$ and $k \geq 1$, we can write

$$P_n(k) = \frac{\binom{n}{k}}{2^n} = \frac{n+1-r}{r}\binom{n}{r-1}.$$

So;

$$P_n(k) = \frac{n+1-r}{r} P_{n-1}. \tag{2.1}$$

Why Recursive Relations?

As opposed complex explicit expressions, concentrating on deriving simple recursive relations much more effective. In this study, all obtained relations are linear, so that they have low complexities. It is feasible to get the exact probabilities for considerably long sequences such as 2^{14} bits by employing these relations.

For example, for the frequency test if we do not have the recursion above we have to find k from this equation:

$$\sum_{i=0}^{k} \binom{n}{i} \frac{1}{2^{n+1}} = \alpha.$$

For the sequences that have long length,for example 4096, it is impossible to find the value of k.

Here are some test results:

α	worst $\alpha - set$	actual α
0.01	$(0, 1965) \cup (2131, 4096)$	$0,009925289$
0.02	$(0, 1973) \cup (2123, 4096)$	$0,01989435$
0.04	$(0, 1982) \cup (2115, 4096)$	$0,03917112$

Table 2.1: Weight Test for n=4096

α	worst $\alpha - set$	actual α
0.01	$(0, 48) \cup (79, 128)$	$0,008007673$
0.02	$(0, 50) \cup (78, 4096)$	$0,016670734$
0.04	$(0, 51) \cup (76, 4096)$	$0,034186728$

Table 2.2: Weight Test for n=128

2.2 Runs Test

A run is defined as an uninterrupted sequence of identical bits[2]. For instance, for a given sequence σ = 00101110010110, there are 9 runs: 00-1-0-111-00-1-0-11-0.
In this study, we take a family of test functions depending on the probabilities defined as below:

- probability that a random sequence of length n has r runs,
- probability that a random sequence of length n has exactly k runs of length a.

where n,r,a are positive integers and k is a nonnegative integer.

A composition of an integer n is a way of writing n as the sum of a sequence of positive integers, concerning the ordering of summands. Given a binary string of length n with r runs, suppose that lengths of the runs are $\ell_1, ..., \ell_r$. Obviously, sum of these lengths is equal to the length of the sequence; therefore, the ordered array $(\ell_1, ..., \ell_r)$ is a composition of n. Hence, lengths of runs of a binary sequence of length n matches a unique composition of n. Contrarily, it matches two sequences (one starting with a one; the other starting with a zero) for each composition of n, so lengths of runs are equal to the corresponding parts of the composition. Any problem on compositions can be envisioned as a problem on the number of runs of a binary sequence because of this 2-1 correspondence. Now we approach compositions for required computations.

In this section, all theorems are taken from [15].

2.2.1 Generating Functions

The generating function for the infinite sequences $\langle g_0, g_1, g_2, ... \rangle$ is the power series:

$$G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + ...$$

A generating function is a formal power series, in which we usually take x as a placeholder instead of a number. A generating function is rarely regarded by letting x take a real number value. Hence, the issue of convergence is generally bypassed by the author. In this way, we can figure out "correspondence" between a sequence and its generating function with a doublesided "arrow" below:

$$\langle g_0, g_1, g_3, ... \rangle \longleftrightarrow g_0 + g_1 x + g_2 x^2 + g_3 x^3 + ...$$

For example, you can see some sequences and their generating functions below:

$$\langle 0, 0, 0, 0, ... \rangle \longleftrightarrow 0 + 0x + 0x^2 + 0x^3 + ... = 0$$
$$\langle 0, 0, 1, 0, ... \rangle \longleftrightarrow 0 + 0x + 1x^2 + 0x^3 + ... = x^2$$
$$\langle 4, 7, 6, 0, ... \rangle \longleftrightarrow 4 + 7x + 6x^2 + 0x^3 + ... = 4 + 7x + 6x^2$$

The pattern here is simple: the ith term in the sequence (indexing from 0) is the coefficient of x^i in the generating function.
The sum of an infinite geometric series is:

$$1 + z + z^2 + z^3 + ... = \frac{1}{1 - z}.$$

This equation does not hold when $|z| \geq 1$ but as described, it is not necessary to worry about convergence issues. This shows closedform generating functions for a whole range of sequences. For instance:

$$\begin{aligned}
\langle 1,1,1,1,...\rangle &\longleftrightarrow 1+1x+1x^2+1x^3+... &&= \frac{1}{1-x}\\
\langle 1,-1,1,-1,...\rangle &\longleftrightarrow 1-x+1x^2-x^3+... &&= \frac{1}{1+x}\\
\left\langle 1,a,a^2,a^3,...\right\rangle &\longleftrightarrow 1+ax+a^2x^2+a^3x^3+... &&= \frac{1}{1-ax}\\
\langle 1,0,1,0,...\rangle &\longleftrightarrow 1+x^2+x^4+... &&= \frac{1}{1-x^2}.
\end{aligned}$$

2.2.2 Using Generating Function on Runs Test

Let n, r be positive integers, recall that, we approve the formal sum $1+z+z^2+...$ correspond to each summand (x_i) in order to find the number of nonnegative integer solutions of the equation $x_1+x_2+...+x_r = n$ and then find the coefficient of z^n in $(1+z+z^2+...)^r$. Any limitation on summands has a natural reflection to the corresponding factors. For example, the number of solutions of $x_1+x_2+...+x_8 = 20$ based on the condition $3 \leq x_i \leq 10$ is given by the coefficient of z^{20} in $(z^3+z^4+...+z^{10})^8$. In a same way, coefficient of z^{10} in $(1+z^2+z^4+...)^4(z+z^3+z^5+...)^4$ is the number of solutions of the same equation if we depend upon exactly half of the summands be even.
If we remember compositions again, we observe that $c(n,r)$, the number of compositions with $r \geq 1$ parts of the positive integer n, is the number of positive integer solutions of the equation

$$x_1+x_2+...+x_r = n \tag{2.2}$$

which is the coefficient of z^n in $(z+z^2+...)^r = \left(\frac{z}{1-z}\right)^r$ or equivalently, $c(n,r)$ is the coefficient of $z^n x^r$ in

$$1+\sum_{r=1}^{\infty}\left(\frac{z}{1-z}\right)^r x^r = \frac{1}{1-(\frac{z}{1-z})x}$$

which means that

$$\mathfrak{C}(z,x) = \frac{1-z}{1-z-zx}.$$

On the other hand, as

$$\frac{z^r}{(1-z)^r} = z^r\sum_{i=0}^{\infty}\binom{-r}{i}(-z)^i = z^r\sum_{i=0}^{\infty}\binom{r+i-1}{r-1}(-z)^i,$$

coefficient of z^n in $(z+z^2+\ldots)^r$ is

$$c(n,r) = \binom{n-1}{r-1}.$$

By definition, $c(n) = \sum_{r=1}^{n} c(n,r)$. In other words, $c(n)$ is the sum of coefficients of z^n in $1, \frac{z}{1-z}, \frac{z^2}{(1-z)^2}, ...$ or equivalently, $c(n)$ is the coefficient of z^n in

$$1 + \sum_{r=1}^{\infty} (\frac{z}{1-z})^r = \frac{1}{1 - \frac{z}{1-z}} = \frac{1-z}{1-2z}.$$

It follows that

$$\mathfrak{C}(z) = \frac{1-z}{1-2z}$$

and consequently, $c(n) = 2c(n-1)$ for $n \geq 2$ and by iteration we get

$$c(n) = 2^{n-1}.$$

2.2.3 Number of Total Runs Test

As stated in [15], the set of all compositions of n is equivalent to the set of all compositions of all integers less than n, including 0, hence for any integer $n \geq 1$

$$c(n) = c(n-1) + \ldots + c(1) + c(0) = \sum_{i=1}^{n} c(n-i). \tag{2.3}$$

Theorem 2.1 ([15]). *The sequence $\{c(n)\}_{n=0}^{\infty}$ is determined with the initial conditions $c(0) = c(1) = 1$ and the recurrence relation $c(n) = 2c(n-1)$ for all integers $n \geq 2$.*

Proof. By convention it's obvious that $c(0) = 1$ and $c(1) = 1$. We can write the recursion 2.3 as

$$c(n-1) = \sum_{i=1}^{n-1} c(n-1-i) = \sum_{i=2}^{n} c(n-i) = \left(\sum_{i=1}^{n} c(n-i) \right) - c(n-1)$$

for $n \geq 2$. One can compare it with expression 2.3 and conclude that $c(n) = 2c(n-1)$ for $n \geq 2$. □

2.2.4 Runs of Length r Test

Theorem 2.2 ([15]). *If a is a fixed positive integer then*

$$\mathfrak{C}_a(z,y,x) = \frac{1-z}{1 - z(x+1) + z^a x(1-z)(1-y)}.$$

Proof. $\mathfrak{C}_a(n,k,r)$ is the number of positive solutions of the equation $x_1 + x_2 + ... + x_r = n$ such that $x_{i_1} = ... = x_{i_k} = a$ for some $\{i_1, ..., i_k\} \subset \{1, ...r\}$ and $x_i \neq a$ for $i \notin \{i_1, ...i_k\}$. The subset $\{i_1, ..., i_k\}$ can be chosen in $\binom{r}{k}$ distinct ways and once this

set is determined, the number of solutions of the equation is given by the coefficient of z^n in

$$(z^a)^k((z+z^2+...)-z^a)^{r-k} = (z^a)^k\left(\frac{z}{1-z}-z^a\right)^{r-k}.$$

It follows that $c_a(n,k,r)$ is the coefficient of $z^n y^k x^r$ in

$$\mathfrak{C}_a(z,y,x) = 1+\sum_{k=0}^{\infty}\sum_{r=1}^{\infty}\binom{r}{k}z^{ka}U^{r-k}y^k x^r \tag{2.4}$$

where $U = \frac{z}{1-z} - z^a$. The double sum on the right hand side can be separated for the cases $k=0$ and $k=1,2,...$ to obtain

$$\begin{aligned}\mathfrak{C}_a(z,y,x) &= 1+\sum_{r=1}^{\infty}U^r x^r + \sum_{k=1}^{\infty}z^{ka}y^k x^k\sum_{r=1}^{\infty}\binom{r}{k}U^{r-k}x^{r-k}\\ &= \frac{1}{1-Ux}+\sum_{k=1}^{\infty}z^{ka}y^k x^k\sum_{r=1}^{\infty}\binom{r}{k}(Ux)^{r-k}\\ &= \frac{1}{1-Ux}+\frac{1}{1-Ux}\sum_{k=1}^{\infty}\left(\frac{z^a yx}{1-Ux}\right)^k\\ &= \frac{1}{1-Ux-z^a yx}\end{aligned}$$

and finally by substituting $U = \frac{z}{1-z}+z^a$, we obtain the desired expression. □

Theorem 2.3 ([15]). *If a is a fixed positive integer then*

$$\mathfrak{C}_a(z,y) = \frac{1-z}{1-2z+z^a(1-z)(1-y)}.$$

Proof. From the proof of Theorem 2.2, we see that $c_a(n,k)$ is the coefficient of z_n in $\sum_{r=1}^{\infty}\binom{r}{k}(z^a)^k U^{r-k}$. It follows that $c_a(n,k)$ is the coefficient of $z^n y^k$ in

$$1+\sum_{k=0}^{\infty}\sum_{r=1}^{\infty}\binom{r}{k}z^{ka}U^{r-k}y^k.$$

From 2.4, we observe that this expression is in fact $\mathfrak{C}_a(z,y,1)$, thus $\mathfrak{C}_a(z,y) = \mathfrak{C}_a(z,y,1)$. □

A recursion for $c_a(n,k)$ can be attained through the way analogous to the one used in attaining 2.3. First consider the case $k=0$. we find such compositions of all integers less than n, except $n-a$ by deleting the first part of each composition of n which has no part equal to a. . Thus, the recursion for $c_a(n,0)$ diverges from the recursion for $c(a)$ only by the summand $c(n-a,0)$, that is

$$c_a(n,0) = \sum_{i=1}^{n}c_a(n-i,0) - c_a(n-a,0). \tag{2.5}$$

Delete the first part of a composition of n which has $k > 1$ parts equal to a. If the deleted part is equal to a, then the rest constitute a composition of $n-a$ with $k-1$ parts equal to a. If the deleted term is equal to $i(i \neq a)$, then the rest form a composition of $n - i$ with k parts equal to a. So we can say

$$c_a(n,k) = \sum_{i=1}^{n} c_a(n-i,k) - c_a(n-a,k) + c_a(n-a,k-1). \tag{2.6}$$

Theorem 2.4 ([15]). *Let a be a positive integer. The sequence $\{c_a(n,k)\}_{n=0}^{\infty}$ is determined with the initial conditions for $k = 0$*

$$c_a(n,0) = \begin{cases} 1 & if\ n = 0 \\ 2^n & if\ a > 1\ and\ 1 \leq n \leq a-1 \\ 2^{a-1} - 1 & if\ n = a \end{cases}$$

and for $k \geq 1$

$$c_a(n,k) = \begin{cases} 0 & if\ n \leq ka-1 \\ 1 & if\ n = ka \end{cases}$$

and the recurrence relations

$$c_a(n,0) = 2c_a(n-1,0) - c_a(n-a,0) + c_a(n-1-a,0) \tag{2.7}$$

for $n \geq a+1$ and

$$c_a(n,k) = 2c_a(n-1,k) - c_a(n-a,k) + c_a(n-a-1,k) + c_a(n-a,k-1) - c_a(n-a-1,k-1) \tag{2.8}$$

for $k \geq 1$ and $n \geq ka+1$.

Proof. When $k = 0$, we have $c_a(0,0) = 0$ by convention. If $n < a$, then no composition of n contains a as a part, so $c_a(n,0) = 2^{n-1}$ for $n \leq a$. Only one composition of a contains a, thus $c_a(a,0) = 2^{a-1} - 1$.

When $k \geq 1$, then $c(0,k) = 0$ by convention. If $n < ka$, then no composition of n can contain k parts equal to a, so $c_a(n,k) = 0$ for $n < ka$. Only one composition of $n = ka$ consists of k parts, each equal to a, thus $c_a(a,k) = 1$.

For $n \geq a+1$, recursion 2.5 can be written as

$$\begin{aligned} c_a(n-1,0) &= \sum_{i=1}^{n-1} c_a(n-1-i,0) - c_a(n-1-a,0) \\ &= \sum_{i=2}^{n} c_a(n-i,0) - c_a(n-1-a,0) \\ &= \sum_{i=1}^{n} c_a(n-i,0) - c_a(n-1,0) - c_a(n-1-a,0). \end{aligned}$$

Comparing this expression to 2.5, we find 2.7.

In a similar way, for $n \geq ka + 1$ we write the recursion 2.6

$$\begin{aligned} c_a(n-1,k) &= \sum_{i=1}^{n-1} c_a(n-1-i,k) - c_a(n-1-a,k) + c_a(n-1-a,k-1) \\ &= \sum_{i=2}^{n} c_a(n-i,k) - c_a(n-1-a,k) + c_a(n-1-a,k-1) \\ &= \sum_{i=1}^{n} c_a(n-i,k) - c_a(n-1,k) - c_a(n-1-a,k) + c_a(n-1-a,k-1) \end{aligned}$$

Comparing this expression to 2.6, we find 2.8. □

2.2.5 Probability Distribution Function of Runs Test

In this section we compute the basic probabilities based on our tests. Let Ω_n be the set of binary sequences of length n and define the following nonnegative integer valued random variables on Ω_n:

$$\begin{aligned} X(\sigma) &= \text{number of runs of } \sigma, \\ X_a(\sigma) &= \text{number of runs of length } a \text{ of } \sigma \end{aligned}$$

we denote the probability mass functions of these random variables as

$$\begin{aligned} p(n,r) &= probability(X = r), \\ p_a(n,k) &= probability(X_a = r). \end{aligned}$$

we have stated that it matches unique composition of n for each binary sequence of length n and it matches exactly 2 binary sequences of length 2 for each composition of n. Then, the number of binary sequences of length n which have r runs is twice the number of compositions of n with r parts, as an instance. Because the number of all binary sequences of length n is 2^n, we exactly get

$$p(n,r) = \frac{1}{2^n}\left(2c(n,r)\right) = 2^{1-n}c(n,r)$$

and

$$p_a(n,k) = 2^{1-n}c_a(n,k).$$

Theorem 2.5 ([15]). *Let a be a positive integer. The sequence $\{p(n,r)\}_{n=0}^{\infty}$ is determined with the initial conditions*

$$p(n,r) = \begin{cases} 1 & if\ r = n \\ 0 & if\ r = 0\ and\ n > 0 \\ 0 & if\ r > 0\ and\ r > n \end{cases}$$

the recursion

$$p(n,r) = \frac{1}{2}\left(p(n-1,r) + p(n-1,r-1)\right)$$

and for $n > r > 0$.

Proof. We already know that $c(n,r) = \binom{n-1}{r-1}$. Pascal's identity $\binom{n-1}{r-1} = \binom{n-2}{r-1} + \binom{n-2}{r-2}$ eads the desired expression. □

Theorem 2.6 ([15])**.** *Let* a *be a positive integer. The sequence* $\{p_a(n,k)\}_{n=0}^{\infty}$ *is determined with the initial conditions:*
for $k = 0$

$$p_a(n,0) = \begin{cases} 1 & if \quad n \leq a-1 \\ 1 - 2^{1-a} & if \quad n \in \{a, a+1\} \end{cases}$$

for $k \geq 1$

$$p_a(n,k) = \begin{cases} 0 & if \quad n \leq ka-1 \\ 2^{1-a} & if \quad n \in \{ka, ka+1\} \end{cases}$$

apart from $p_1(2,0) = 1/2$ *and* $p_1(2,1) = 0$.

The recurrence relations

- *for* $n \geq a+2$

$$p_a(n,0) = p_a(n-1,0) - 2^{-a}p_a(n-a,0) + 2^{-a-1}p_a(n-1-a,0)$$

- *for* $r \geq 1$ *and* $n \geq ra+2$

$$\begin{aligned} p_a(n,k) = p_a(n-1,k) - 2^{-a}p_a(n-a,k) + 2^{-a-1}p_a(n-a-1,k) \\ + 2^{-a}p_a(n-a,r-1) - 2^{-a-1}p_a(n-a-1,r-1). \end{aligned}$$

Proof. By convention, $p_a(0,0) = 1$. Initial conditions can be controlled by direct computing. For the other cases, just substitute $p_a(n,k) = 2^{1-n}c_a(n,k)$ in 2.4. □

One can see some test results below:

α	worst $\alpha - set$	actual α
0.01	$(0,1966) \cup (2132,4096)$	$0,009925289$
0.02	$(0,1974) \cup (2124,4096)$	$0,019894351$
0.04	$(0,1982) \cup (2115,4096)$	$0,039123224$

Table 2.3: Number of Runs Test for n=4096

α	worst $\alpha - set$	actual α
0.01	$(0, 49) \cup (80, 128)$	$0,007519595$
0.02	$(0, 51) \cup (79, 128)$	$0,016670734$
0.04	$(0, 52) \cup (77, 128)$	$0,032789562$

Table 2.4: Number of Runs Test for n=128

α	worst $\alpha - set$	actual α
0.01	$(0, 932) \cup (1118, 4096)$	$0,009714587$
0.02	$(0, 941) \cup (1109, 4096)$	$0,019586516$
0.04	$(0, 951) \cup (1099, 4096)$	$0,039928555$

Table 2.5: Number of Runs of length 1 Test for n=4096

α	worst $\alpha - set$	actual α
0.01	$(0, 16) \cup (50, 128)$	$0,008763745$
0.02	$(0, 18) \cup (49, 128)$	$0,017565686$
0.04	$(0, 19) \cup (46, 128)$	$0,039504471$

Table 2.6: Number of Runs of length 1 Test for n=128

α	worst $\alpha - set$	actual α
0.01	$(0, 457) \cup (568, 4096)$	$0,009335232$
0.02	$(0, 462) \cup (562, 4096)$	$0,019336163$
0.04	$(0, 468) \cup (556, 4096)$	$0,039830136$

Table 2.7: Number of Runs of length 2 Test for n=4096

α	worst $\alpha - set$	actual α
0.01	$(0, 6) \cup (27, 64)$	$0,006718844$
0.02	$(0, 7) \cup (26, 64)$	$0,01501039$
0.04	$(0, 8) \cup (25, 64)$	$0,031320682$

Table 2.8: Number of Runs of length 2 Test for n=128

2.3 Random Walk Excursion Test

Let σ be a binary string of length n denoted by $s_1 s_2 ... s_n$. We call a string *balanced*, if it has an equal number of 1's and 0's. Say s_k is a *balanced point* if substring of σ $s_1 s_2 ... s_k$ is balanced.
A string σ is said to intersect the line $y = t$ as s_i if $2(s_1 + s_2 + ... + s_i) - i = t$ for $i = 1, ..., n$. We can see that obviously s_i is a *balanced point* if and only if σ intersects the line $y = 0$ at s_i.
Say $X_t(n, k)$ be the set of strings of length n intersecting the line $y = t$ exactly at k distinct terms and let $x_t(n, k) = |X_t(n, k)|$. We can say that from the definition $x_0(n, k) = x(n, k)$ and it is obvious that $x_t(n, k) = x_{-t}(n, k)$ for any $t = 1, ..., n$.
$B(n, k)$ stands for the set of balanced strings containing exactly k balance points and

$b(n,k)$ is the number of such strings. $B_t(n)$ denotes the set of strings of length n which touch the line $y=t$ for the first time at the last term and $b_t(n)=|B_t(n)|$. You can see that $B_0(n)=B(n,1)$. Also, no string in $B_t(n)$ is balanced if $t\neq 0$.
$X(n,k)$ stands for the set of strings containing exactly k balance points and $x(n,k)$ is the number of such strings. As previously, for $t=0$, we can write $X_t(n)=X_t(n,0)$ and $x_t(n)=|X_t(n,0)|$.
While $X_t(n)$ is the set of strings of length n which do not intersect the line $y=t$, complement of this set which we can call $\overline{X}_t(n)$ is the set of strings which intersects the line $y=t$ at least in one point. So, $\overline{x}_t(n)=\left|\overline{X}_t(n)\right|$. The probability of a string of length n to have k intersections with the line $y=t(ory=-t)$ is denoted by $p_t(n,k)$. Lastly, $[a(n,k)]$ denotes the table(or matrix) whose rows are indexed by $i=1,...,n$ and columns are indexed by $j=1,...k$ where n and k are positive inetegers. Likewise, $a(i,j)$ denotes two dimensional array for $i=1,...,n$ and $j=1,...k$.

In this section, all theorems are taken by [3].

2.3.1 Catalan Numbers

One of the basic tools used here is the sequence $\{\mathcal{C}\}_{n_0}^{\infty}$ of Catalan numbers where

$$\mathcal{C}_n=\frac{1}{n+1}\binom{2n}{n}$$

for any nonnegative integer n. First a few terms of this sequence are 1,1,2,5,14,...

It is sincere to see that Catalan numbers satisfy the following recursion for $n>1$

$$\mathcal{C}_n=\frac{4(n-1)}{n+1}\mathcal{C}_{n-1}. \tag{2.9}$$

Another important property of Catalan numbers is that, convolution of the sequence $\mathcal{C}_{n_0}^{\infty}$ is itself, that is, for any nonnegative integer n,

$$\mathcal{C}_n=\sum_{i=0}^{n-1}\mathcal{C}_i\mathcal{C}_{n-i}.$$

Generating function of this sequence is

$$\mathcal{C}(z)=\sum_{i=0}^{\infty}\mathcal{C}_i z^i=1+z+2z^2+5z^3+14z^4+\ldots.$$

Using this property it is easy to verify that

$$z\mathcal{C}^2(z)=\mathcal{C}(z)-1. \tag{2.10}$$

By differentiating both sides of 2.10 one obtains

$$\frac{d}{dz}\mathcal{C}(z)=\frac{\mathcal{C}^2(z)}{1-2z\mathcal{C}}$$

and by differentiating the product $z\mathcal{C}(z) = \sum_{i=0}^{\infty} \frac{1}{i+1}\binom{2i}{i} z^{i+1}$ we obtain the generating function of the sequence $\left\{\binom{2n}{n}\right\}_n$:

$$\mathcal{C}(z) + z\frac{d}{dz}\left(\mathcal{C}(z)\right) = \sum_{i=0}^{\infty} \binom{2i}{i} z^i. \tag{2.11}$$

Following lemma presents a result which will be the basis of many computations throughout the work.

Lemma 2.7 ([3]). *Let n,t and q be positive integers with $t \leq q \leq n$. The number of strings of length n which contain q zeros and which intersect the line $y = t$ at least once is given by*

$$\begin{cases} \binom{n}{q-t} & \text{if} \quad t \leq q \leq \frac{n+t}{2} \\ \binom{n}{q} & \text{if} \quad \frac{n+t}{2} \leq q \leq n \end{cases} \tag{2.12}$$

Proof. Given a string σ of length n which intersects the line $y = t$, depending on q we consider two cases:

- In $\frac{n+t}{2} \leq q \leq n$ case σ necessarily intersects the line $y = t$ and number of such strings is
$$\binom{n}{q}.$$
- Let us look at $t \leq q \leq \frac{n+t}{2}$ case. Let A be the set of strings of length n which have q zeros and which intersect the line $y = t$, ant let B be the set of strings of length n which have $q-t$ zeros. We will present that these two sets are equivalent which is why the number of strings in A is
$$\binom{n}{q-t}.$$
Given $\sigma \in A$. Let i_0 be the smallest integer such that σ intersects the line $y = t$ at s_{i_0}. The string $\overline{\sigma} = \overline{s_1} \ldots \overline{s_{i_0}} s_{i_0+1} \ldots s_n$ where $\overline{s_i} = 1 - s_i$, $i = 1, ..., i_0$ has $q - t$ zeros, hence $\sigma \in B$. Hence, it matches a unique string $\overline{\sigma} \in B$ for each $\sigma \in A$. Contrarily, any string τ in B has $q - t$ zeros, hence $n - q + t$ ones. However, the condition $q \leq (n+t)/2$ shows that $n-q+t \geq (n+t)/2$, meaning the string τ intersects the line $y = -t$. Now in the string τ, starting with the first term replace each one with a zero and each zero with a one up to the term at which the string intersects the line $y = -t$ for the first time. The resulting string intersects the line $y = t$ and has q zeros, hence is in A. Then the correspondence given above is one to one and the sets A and B are equivalent.

□

Lemma 2.8 ([3]). *Let n and t be positive integers with $t \leq n$. The number of strings of length n which intersect the line $y = t$ at least once is given by*

$$\overline{x}_t(n) = \begin{cases} 2\sum_{i=0}^{\frac{n-t}{2}} \binom{n}{i} - \binom{n}{\frac{n-t}{2}} & \text{if } n+t \text{ is even} \\ 2\sum_{i=0}^{\frac{n-t-1}{2}} \binom{n}{i} & \text{if } n+t \text{ is odd} \end{cases} \tag{2.13}$$

Proof. Depending on the parity of $n+t$, we examine two cases separately:

- $n+t$ is even

$$\begin{aligned}\overline{x}_t(n) &= \sum_{i=t}^{\frac{n+t}{2}-1} \binom{n}{i-t} + \sum_{i=\frac{n+t}{2}}^{n} \binom{n}{i} \\ &= \sum_{i=0}^{\frac{n-t}{2}-1} \binom{n}{i} + \sum_{i=0}^{\frac{n-t}{2}} \binom{n}{i} \\ &= 2\sum_{i=0}^{\frac{n-t}{2}} \binom{n}{i} - \binom{n}{\frac{n-t}{2}}\end{aligned}$$

- $n+t$ is odd

$$\begin{aligned}\overline{x}_t(n) &= \sum_{i=t}^{\frac{n+t-1}{2}} \binom{n}{i-t} + \sum_{i=\frac{n+t+1}{2}}^{n} \binom{n}{i} \\ &= \sum_{i=0}^{\frac{n-t-1}{2}} \binom{n}{i} + \sum_{i=0}^{\frac{n-t-1}{2}} \binom{n}{i} \\ &= 2\sum_{i=0}^{\frac{n-t-1}{2}} \binom{n}{i}\end{aligned}$$

□

2.3.2 Recursive Relations Satisfied by $b(n,k)$

We first define $B(n,1)$ as the number of balanced sequences having no balance points other than the last term. It is clear that a balanced sequence must be of even length, so $B(n,1) = 0$ for any odd integer n. We have the following proposition for sequences of even length, as below:

Proposition 2.9 ([3])**.** *For any positive integer m,*

$$b(2m,1) = 2\mathcal{C}_{2m-1}$$

where $\mathcal{C}_{m-1}$ is a Catalan number.

Proof. Any $\sigma = s_1 \cdots s_{2m} \in B(2m,1)$ is balanced and has only one balance point (necessarily the last term) and none of the terms $s_1, \cdots, s_{2m-1}$ is a balance point. For $m = 1$ the claim is clear: $b(2,1) = 2 = 2\mathcal{C}_0$. Now assume that $m > 1$ and $s_1 = 1$ (hence, $s_{2m} = 0$). It is easy to observe that the string $s_2, \cdots, s_{2m-1}$ is balanced

and it cannot intersect the line $y = 1$. Therefore, there corresponds a unique string $\sigma \in B(2m, 1)$ with $s_1 = 1$ for each such string. The number of strings in $B(2m, 1)$ is equal to the number of strings of length $2m - 2$ having $q = m - 1$ zeros and not intersecting the line $y = 1$ because the converse relation also holds. After that, from 2.12 we obtain $B(2m, 1) = \binom{2m-2}{m-1} - \binom{2m-2}{m-2}$ which simplifies into the $\mathcal{C}_{m-1}$. By including the strings with initial term 0, the assertion follows. □

In conclusion, for any nonnegative integer we have

$$b(n,1) = \begin{cases} 0 & \text{if } n = 0 \text{ or n is odd} \\ 2\mathcal{C}_{\frac{n}{2}-1} & \text{if } n > 0 \text{is even} \end{cases} \tag{2.14}$$

Proposition 2.10 ([3]). *For any positive integers m and $k > 1$, the sequence $\{b(2m, k)\}_{n=0}^{\infty}$ is convolution of the sequences $\{b(n, 1)\}_{n=0}^{\infty}$ and $\{b(2m, k-1)\}_{n=0}^{\infty}$, that is*

$$b(2m,k) = \sum_{i=0}^{m-1} b(2i,1)b(2m-2i,k-1).$$

Proof. Let $k > 1$ and consider a string $\sigma \in B(n, k)$. Assume that the first balance point is s_{2i}. Then, σ can be divided into two substrings $\sigma_1 = s_1 \cdots s_{2i}$ and $\sigma_2 = s_{2i+1} \cdots s_{2m}$ such that $\sigma_1 \in B(2i, 1)$ and $\sigma_2 \in B(2m - 2i, k - 1)$. □

Expression 2.14 provides us compute the first row of $[b(n, k)]$ by $n/2$ multiplications. Then, using above proposition, for each $k > 1$, computation of terms on kth column requires $\frac{(n-2k)(n-2k+2)}{8}$ multiplications and one less additions. As a result, the total number of multiplications and additions for computing the entire table $[b(n, k)]$ are $\frac{n^3-n}{6}$ and $\frac{n^3-4n}{6}$, respectively.

Now we point out generating function of the sequence $\{b(n, k)\}_{n=0}^{\infty}$. First we find the generating function $\mathcal{B}(z)$ of $\{b(n, 1)\}_{n=0}^{\infty}$:

$$\begin{aligned} \mathcal{B}(z) &= \sum_{i=0}^{\infty} b(i,1)z^i \\ &= \sum_{i=1}^{\infty} b(i,1)z^{2i} \\ &= 2\sum_{i=1}^{\infty} \mathcal{C}_{i-1}z^{2i} \\ &= 2z^2 \sum_{i=0}^{\infty} \mathcal{C}_i z^{2i} \\ &= 2z^2\mathcal{C}(z^2). \end{aligned}$$

Proposition 2.11 ([3]). *Let k be a positive integer. Then generating function of the sequence $\{b(n,k)\}_{n=0}^{\infty}$ is*

$$\mathcal{B}^k(z) = 2^k z^{2k} \mathcal{C}^k(z^2).$$

Proof. Proposition 2.10 shows that the generating function of $\{b(n,k)\}_{n=0}^{\infty}$ is the product of $\mathcal{B}(z)$ and the generating function of $\{b(n,k-1)\}_{n=0}^{\infty}$. Then, proof follows inductively: generating function of $\{b(n,k)\}_{n=0}^{\infty}$ for $k=2$ is $\mathcal{B}(z)\mathcal{B}(z) = \mathcal{B}^2(z)$. For $k=3$ we have $\mathcal{B}(z)\mathcal{B}^2(z) = \mathcal{B}^3(z)$ and so on. □

Theorem 2.12 ([3]). *For any positive integers n and k, the quantities $b(n,k)$ satisfy the following recursions with regard to the given initial conditions.*

Proof.

- for $k=1$

$$b(n,1) = \begin{cases} 0 & \text{if } n=1 \\ 2 & \text{if } n=2 \\ \dfrac{4(n-3)}{n} b(n-2,1) & \text{if } n \geq 3 \end{cases},$$

- for $k=2$

$$b(n,2) = \begin{cases} 0 & \text{if } n \leq 2 \\ 2b(n,1) & \text{if } n \geq 3 \end{cases},$$

- for $k \geq 3$

$$b(n,k) = \begin{cases} 0 & \text{if } n < 2k \\ 2b(n,k-1) - 4b(n-2,k-2) & \text{if } n \geq 2k \end{cases}.$$

- Initial terms are obvious and the recursion follows from 2.9 and 2.12.

- Initial terms are obvious. Generating function of $\{b(n,2)\}_n$ satisfies.

$$\begin{aligned} \mathcal{B}^2(z) &= 4x^4 \mathcal{C}^2 z^2 \\ &= 4z^2[z^2 \mathcal{C}^2(z^2)] \\ &= 4z^2[\mathcal{C}(z^2) - 1] \\ &= 4z^2 \mathcal{C}(z^2) - 4z^2 \\ &= 2\mathcal{B}(z) - 4z^2 \end{aligned}$$

which means that $b(2,2) = 2b(2,1) - 4 = 0$ and for $n > 2$, $b(n,2) = 2b(n,1)$.

- Initial terms are obvious. For any integer $k > 2$ we have

$$\begin{aligned}\mathcal{B}^k(z) &= 2^k z^{2k}\mathcal{C}^k(z^2)\\ &= 2^k z^{2k-2}\mathcal{C}^{k-2}[z^2\mathcal{C}^2(z^2)]\\ &= 2^k z^{2k-2}\mathcal{C}^{k-2}[\mathcal{C}^2(z^2)-1]\\ &= 2[2^{k-1}z^{2k-2}\mathcal{C}^{k-1}(z^2)] - 4z^2[2^{k-2}z^{2k-4}\mathcal{C}^{k-2}(z^2)]\\ &= 2\mathcal{B}^{k-1}(z) - 4z^2\mathcal{B}^{k-2}(z)\end{aligned}$$

  which implies that $b(n,k) = 2b(n,k-1) - 4b(n-2,k-2)$ for any integer $n > 2$.

□

Theorem 2.12 is important that it diminishes the complexity of computation of $[b(n,k)]$ as follows. We can compute the first row by $n/2$ multiplications by first part of the theorem. Starting from the third row, each term can be computed by 2 multiplications and 1 additions, then we need a total of $\frac{n^2+2n}{4}$ multiplications and $\frac{n^2-2n}{8}$ additions for the entire table.

2.3.3 Recursive Relations Satisfied by $x(n,k)$

Given a positive integer n, by substituting $t = 1$ in 2.13 we see that

$$\bar{x_1}(n) = \begin{cases} 2^n - \binom{n}{\frac{n-1}{2}} & \text{if } n \text{ is odd} \\ 2^n - \binom{n}{\frac{n}{2}} & \text{if } n \text{ is even} \end{cases}$$

which can be written simply as $\bar{x_1}(n) = 2^n - \binom{n}{\lfloor \frac{n}{2} \rfloor}$. On the other hand, by definition, $x_1(n) = 2^n - \bar{x_t}(n)$ which gives the number of strings which do not intersect the line $y = 1$ as

$$x_1(n,0) = x_1(n) = \binom{n}{\lfloor \frac{n}{2} \rfloor}. \tag{2.15}$$

Now, let $\sigma \in X_0(n)$ and assume that $s_1 = 1$, then $s_2 \cdots s_n \in X_1(n-1,0)$. It follows that the number of strings in $X_0(n)$ with the first term 0 is $X_1(n-1,0) = \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$. Because the same holds for the strings with the first term 0, we get the number of strings which do not intersect the line $y = 0$ as

$$x_0(n,0) = x_0(n) = \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}. \tag{2.16}$$

Let $\mathcal{X}_k(z)$ be the generating function of the sequence $\{x(n,k)\}_{n=0}^{\infty}$ and for the special case $k=0$ write $\mathcal{X}(z)=\mathcal{X}_0(z)$. We have $\mathcal{X}(z)=\sum_{i=0}^{\infty}x(i,0)z^i$, where we let $x(0,0)=1$. We can write this function as $\mathcal{X}(z)=\sum_{i=0}^{\infty}x(2i,0)z^{2n}+x(2i+1,0)z^{2n+1}$. From 2.13 we obtain $x(2i,0)=2\binom{2i-1}{i}=\binom{2i}{i}$ and $x(2i+1,0)=2\binom{2i}{i}$, thus

$$\mathcal{X}(z)=\sum_{i=0}^{\infty}\left(\binom{2i}{i}+2\binom{2i}{i}z\right)z^{2n}=(1+2z)\sum_{i=0}^{\infty}\binom{2i}{i}z^{2i}.$$

Now, from 2.11 $\sum_{i=1}^{\infty}\binom{2i}{i}z^{2i}=\mathcal{C}(z^2)+z^2\mathcal{C}'(z^2)$ which leads to

$$\mathcal{X}(z)=(1+2z)(\mathcal{C}(z^2)+z^2\mathcal{C}'(z^2)).$$

Proposition 2.13 ([3]). *For any positive integer n*

$$x(n,0)=\binom{n-1}{\lfloor\frac{n-1}{2}\rfloor}.$$

and for any integer $k>1$,

$$x(n,k)=\sum_{i=1}^{\lfloor n/2\rfloor}b(2i,k-1)x(n-2i,0).$$

Proof. Let $k>1$ and consider a string $\sigma\in X(n,k)$. Assume that the last balance point is s_{2i}. Then, σ can be separated into two substrings $\sigma_1=s_1\cdots s_{2i}$ and $\sigma_2=s_{2i+1}\cdots c_n$ such that $\sigma_1\in B(2i,k)$ and $\sigma_2\in B(n-2i,k-1)$. □

Proposition 2.14 ([3]). *For any positive integer k, generating function of the sequence $\{x(n,k)\}_{n=0}^{\infty}$ is*

$$\mathcal{X}_k(z)=\mathcal{X}(z)B^k(z).$$

Proof. Previous proposition implies that $\mathcal{X}_k(z)=\mathcal{X}_{k-1}(z)B(z)$. Then for $\mathcal{X}_1(z)=\mathcal{X}(z)B(z)$ and assertion follows inductively. □

With the notation of above proposition, if we substitute $k=0$, we see that $\mathcal{X}_0(z)=\mathcal{X}(z)\mathcal{B}^0(z)=\mathcal{X}(z)$.

Theorem 2.15 ([3]). *For any nonnegative integers n and k, the quantities $x(n,k)$ satisfy the following recursions subject to the given initial conditions.*

- *for $k=0$*

$$x(n,0)=\begin{cases}1 & \text{if } n=0\\ 2 & \text{if } n=1\\ 2\left(1-\frac{1}{n}\right)x(n-1,0) & \text{if } n\geq 2 \text{ is even}\\ 2x(n-1,0) & \text{if } n\geq 3 \text{ is odd}\end{cases}$$

- *for* $k = 1$

$$x(n,1) = \begin{cases} 0 & \text{if } n \leq 1 \\ x(n,0) & \text{if } n \geq 2 \end{cases}$$

- *for* $k \geq 2$

$$x(n,k) = \begin{cases} 0 & \text{if } n < 2k \\ 2x(n,k-1) - 4x(n-2,k-2) & \text{if } n \geq 2k \end{cases}.$$

Proof. In all parts, initial conditions follow directly from the definitions.

- If n is odd, say $n = 2m+1$ we have

$$x(2m+1,0) = 2\binom{2m}{m} = 4\binom{2m-1}{m-1} = 2x(2m,0),$$

and if n is even, say $n = 2m$, then

$$x(2m,0) = 2\binom{2m-1}{m-1} = 2\frac{2m-1}{m}\binom{2m-2}{m-1} = 2\left(1 - \frac{1}{2m}\right)x(2m-1,0).$$

- Since $x(0,0) = 1$, $x(1,0) = 2$, $x(0,1) = x(1,1) = 0$, it is sufficient to show that $\mathcal{X}_1(z) = \mathcal{X}(z) - 1 - 2z$.

$$\begin{aligned} \mathcal{X}_1(z) &= \mathcal{X}(z)\mathcal{B}(z) \\ &= (1+2z)(2z^2\mathcal{C}^2(z^2) + 2z^4\mathcal{C}(z^2)\mathcal{C}'(z^2)) \end{aligned}$$

From 2.11 we write $2z\mathcal{C}^2(z^2)+4z^3\mathcal{C}(z^2)\mathcal{C}'(z^2) = 2z\mathcal{C}'(z^2)$ which yields $2z^4\mathcal{C}(z^2)\mathcal{C}'(z^2) = z^2\mathcal{C}'(z^2) - z^2\mathcal{C}^2(z^2)$. Substituting this expression in the above equation we get

$$\begin{aligned} \mathcal{X}_1(z) &= (1+2z)(z^2\mathcal{C}'(z^2) + z^2\mathcal{C}^2(z^2)) \\ &= (1+2z)(z^2\mathcal{C}'(z^2) + \mathcal{C}(z^2) - 1) \\ &= \mathcal{X}(z) - 1 - 2z. \end{aligned}$$

- For any integer $k \geq 2$ we have

$$\begin{aligned} \mathcal{X}_k(z) &= \mathcal{X}(z)\mathcal{B}^k(z) \\ &= \mathcal{X}(z)2\mathcal{B}^{k-1}(z) - 4z^2\mathcal{B}^{k-2}(z) \\ &= 2\mathcal{X}_{k-1}(z) - 4z^2\mathcal{X}_{k-2}(z) \end{aligned}$$

which implies that $x(n,k) = 2x(n,k-1) - 4x(n-2,k-2)$.

□

2.3.4 Recursive Relations Satisfied by $x_t(n,k)$

We have defined $X_t(n,k)$ as the set of strings intersecting the line $y=t$ at exactly k terms. For $t=0$ we have already attained recursive relations by which $[x_0(n,k)]$ can be computed effectively. Thus, we focus on the case $t \neq 0$ and now we can assume that t is positive without loss of generality because $x_{-t}(n,k) = x_t(n,k)$.

Proposition 2.16 ([3]). *Given integers $n,k \geq 0$ and $t > 0$. If $n < t+2k-2$, then $x_t(n,k) = 0$. If $n \geq t+2k-2$, then*

$$\begin{aligned} x_1(n,k) &= \frac{1}{2}x(n+1,k), \\ x_2(n,k) &= \begin{cases} x_1(n+1,0) & \text{if } k=0 \\ x_1(n+1,k) - x(n,k-1) & \text{if } k \geq 1 \end{cases}, \\ x_t(n,k) &= x_{t-1}(n+1,k) - x_{t-2}(n,k) \ (t \geq 3) \end{aligned}$$

Proof. Assume that $\sigma = s_1 \cdots s_n \in X(n,k)$. There are two possibilities, either $s_1 = 0$ and $s_2 \cdots s_n \in X_{-1}(n-1,k)$ or $s_1 = 1$ and $s_2 \cdots s_n \in X_1(n-1,k)$ which implies that $x(n,k) = 2x_1(n-1,k)$. Let $\sigma = s_1 \cdots s_{n+1} \in X_0(n+2,0)$. If $s_1 = 0$ (respectively $s_1 = 1$), then necessarily $s_2 = 0$ (respectively $s_2 = 1$). Thus, there are two possibilities either $s_1 = s_2 = 0$ and $s_3 \cdots s_{n+2} \in X_{-2}(n,0)$ or $s_1 = s_2 = 1$ and $s_3 \cdots s_{n+2} \in X_2(n,0)$. Thus $x(n+2,0) = 2x_2(n,0)$, that is

$$x_2(n,0) = \frac{1}{2}x(n+2,0) = x_1(n+1,0).$$

If $k \geq 1$ and $\sigma = s_1 \cdots s_{n+1} \in X_1(n+1,k)$ then there are two possibilities, either $s_1 = 0$ or $s_2 \cdots s_{n+1} \in X_0(n,k-1)$ or $s_1 = 1$ and $s_2 \cdots s_{n+1} \in X_2(n,k)$. Hence $x_1(n+1,k) = x(n,k-1) + x_2(n,k)$.

Finally, if $\sigma = s_1 \cdots s_{n+1} \in X_{t-1}(n+1,k)$ where $t \geq 3$, then there are two possibilities, either $s_1 = 0$ and $s_2 \cdots s_{n+1} \in X_{t-2}(n,k)$ or $s_1 = 1$ and $s_2 \cdots s_{n+1} \in X_t(n,k)$.

Thus $x_{t-1}(n+1,k) = x_{t-2}(n,k) + x_t(n,k)$. □

2.11 provides us compute all matrices $[x_{t_0}(n,k)]$ for $t = 0,1,2,\ldots$ up to t_0, recursively. We now get a recursive relation by which we can compute the table for $t_0 > 0$ without requiring the tables for $0,1,\ldots,t_0-1$, except the column 0, which is obtained from the corresponding column of the table for $t_0 - 1$.

Theorem 2.17 ([3]). *Let $n \geq 0$, $k \geq 0$ and $t \geq 1$ be integers. Quantities $x_t(n,k)$ satisfy the following recursions.*

$$x_t(n,k) = \begin{cases} x_1(n,k) = \frac{1}{2}x(n+1,k) & \text{if } t=1, \\ \frac{1}{2}(x_{t-1}(n+1,0) + x_{t-1}(n+1,1)) & \text{if } t \geq 2 \text{ and } k=0. \\ x_t(n,k) = \frac{1}{2}x_{t-1}(n+1,k+1) & \text{if } t \geq 2 \text{ and } k \geq 1 \end{cases}$$

Proof. First equality is just a repetition of the first part of 2.16.

For the second inequality, since $x(n,0) = x(n,1)$ for $n \geq 2$, from $x_1(n,k) = \frac{1}{2}x(n+1,k)$ we see that $x_1(n,0) = x_1(n,1)$. Then, equality $x_2(n,0) = x_1(n+1,0)$ can be written as

$$x_2(n,0) = \frac{1}{2}(x_1(n+1,0) + x_1(n+1,1)).$$

Now, we continue with induction on t. Let $t_0 \geq 2$ and assume that

$$x_t(n,0) = (x_{t-1}(n+1,0) + x_{t-1}(n+1,1)) \tag{2.17}$$

holds for any $t \geq t_0$, then

$$\begin{aligned} x_{t_0+1}(n,0) &= x_{t_0}(n+1,0) - x_{t_0-1}(n,0) \\ &= \frac{1}{2}(x_{t_0}(n+2,0) - x_{t_0}(n+2,1)) - \frac{1}{2}(x_{t_0}(n+1,0) + x_{t_0}(n+1,1)) \\ &= \frac{1}{2}(x_{t_0}(n+2,0) - x_{t_0}(n+1,0)) + \frac{1}{2}(x_{t_0}(n+2,1) - x_{t_0}(n+1,1)) = \\ &= \frac{1}{2}(x_{t_0-1}(n+1,0) + x_{t_0-1}(n+1,1)) \end{aligned}$$

which implies that 2.17 holds for all positive integers t.

For the last equality, first note that

$$\begin{aligned} x_2(n,k) &= x_1(n+1,k) - x_0(n,k-1) \\ &= x_1(n+1,k) - 2x_1(n-1,k-1) \\ &= \frac{1}{2}x_1(n+1,k+1). \end{aligned}$$

Now, for $k \geq 1$ and for a fixed integer $t_02 \geq 2$ assume that

$$x_t(n,k) = \frac{1}{2}x_{t-1}(n+1,k+1), \tag{2.18}$$

holds for any $t \leq t_0$, then

$$\begin{aligned} x_{t_0+1}(n,k) &= x_{t_0}(n+1,k) - x_{t_0-1}(n,k) \\ &= x_{t_0}(n+1,k) - 2x_{t_0}(n-1,k-1) \\ &= \frac{1}{2}x_{t_0}(n+1,k+1) \end{aligned}$$

which implies that 2.18 holds for all positive integers t. □

2.3.5 Probability Distribution Function of Excursion Test

Theorem 2.18 ([3]). *Let n, k and t be nonnegative integers. The table $[p_t(n,k)]$ can be constructed by the following recursions*

i) For $t = 0$ *and* $k = 0$

$$p_0(n,0) = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ (1 - \frac{1}{n})x_0(n-1,0) & \text{if } n \geq 2 \text{ is even} \\ p_0(n-1,0) & \text{if } n \geq 3 \text{ is odd} \end{cases}$$

ii) For $t = 0$ *and* $k = 1$

$$p_0(n,1) = \begin{cases} 0 & \text{if } n \leq 1 \\ p_0 x(n-1,0) & \text{if } n \geq 2 \end{cases}$$

iii) For $t = 0$ *and* $k \geq 2$

$$p_0(n,k) = \begin{cases} 0 & \text{if } n < 2k \\ 2p_0(n,k-1) - p_0(n-2,k-2) & \text{if } n \geq 2k \end{cases}$$

iv) For $t = 1$

$$p_1(n,k) = p_0(n+1,k)$$

v) For $t \geq 2$ *and* $k = 1$

$$p_t(n,0) = p_{t-1}(n+1,0) + p_{t-1}(n,1)$$

vi) For $t \geq 2$ *and* $k \geq 2$

$$p_t(n,k) = p_{t-1}(n+1,k+1).$$

Proof. Just substitute $p_t(n,k) = 2^{-n}x_t(n,k)$ in Theorem 2.15 and 2.17. □

Here are some test results:

α	worst $\alpha - set$	actual α
0.01	$(162, 2048)$	$0,00979953$
0.02	$(147, 2048)$	$0,0193272$
0.04	$(130, 2048)$	$0,039003159$

Table 2.9: Excursion Test of y=0 line for n=4096

α	worst $\alpha - set$	actual α
0.01	$(27, 64)$	$0,007206215$
0.02	$(24, 64)$	$0,018740915$
0.04	$(22, 64)$	$0,03294666$

Table 2.10: Excursion Test of y=0 line for n=128

α	worst $\alpha - set$	actual α
0.01	$(162, 2048)$	$0,00979953$
0.02	$(147, 2048)$	$0,0193272$
0.04	$(130, 2048)$	$0,039003159$

Table 2.11: Excursion Test of y=1 line for n=4096

α	worst $\alpha - set$	actual α
0.01	$(27, 64)$	$0,007206215$
0.02	$(24, 64)$	$0,018740915$
0.04	$(22, 64)$	$0,03294666$

Table 2.12: Excursion Test of y=1 line for n=128

2.4 *b*-bit Integer Tests

b-bit Integer Tests are constructed for checking integer sequences. Many properties of integer sequences like maximum term, minimum term, distribution and saturation point of integers are tested by b-bit integer tests. These tests obviously can also be applied on binary sequences.

In the integer sequence case, if the integer size is proper for the test, then the sequence is taken as is. If the sequence necessitate a different integer size, then, the sequence is converted first to binary and then to the required integer length-sequence. The binary sequences are directly converted to the integer sequences as follows. Let σ be a binary and b be the required integer size for the test. Then, the converted sequence $\tilde{\sigma}$ will be $\tilde{\sigma} = u_1, u_2, ..., u_l$ where u_i is the integer whose binary representation is the subsequence $s_{(i-1)b+1}s_{(i-1)b+2}...s_ib$, ie

$$u_i = \sum_{j=1}^{b} 2^{b-j} s_{(i-1)b+j}$$

with $l = [n/b]$, $M = 2^b$ and $u_i \in 0, 1, ..., M-1$, $i = 1, ..., l$.
For example if $\sigma = 010100110101$ then $\tilde{\sigma} = (010)_2(100)_2(110)_2(101)_2 = 2, 4, 6, 5$ is the 3-bit representation of σ.

Another question is how to decide which b value is proper for given $n - bit$ sequence. We will give the way of choosing b value separately for every integer tests.

2.4.1 Saturation Point Test

Saturation Point Test is related with Knuth's Coupon Collector test. The subject of Saturation Point Test which is defined by Sulak [12] is the index of integer, denoted by XS, where all possible integers occur in the given sequence.

2.4.1.1 Probability Distribution Function

The first $k-1$ terms of the sequence must cover $M-1$ distinct integers in order to have k as the saturation point, and the k^{th} term must be the missing one in the first $k-1$ terms. Probability of the first $k-1$ terms of the sequence covering $M-1$ distinct integers is

$$P(XS=k) = P(XC=k-1)\frac{1}{M} = M^{-(k-1)}\left\{ {k-1 \atop M-1} \right\}(M-1)! \tag{2.19}$$

2.4.1.2 Recursion

We have $P_1(1)=1$, $P_i(0)=0$ and $P_0(i)=0$ for i=1,2,... and

$$\begin{aligned}
P_M(k) &= \frac{(M-1)!}{M^{k-1}}\left\{ {k-1 \atop M-1} \right\} \\
&= \frac{(M-1)!}{M^{k-1}}\left[\left\{ {k-2 \atop M-2} \right\} + (M-1)\left\{ {k-2 \atop M-1} \right\}\right] \\
&= \frac{(M-1)!}{M^{k-1}}\left\{ {k-2 \atop M-2} \right\} + \frac{(M-1)!(M-1)}{M^{k-1}}\left\{ {k-2 \atop M-1} \right\} \\
&= \left(\frac{M-1}{M}\right)^{k-1}\left[\frac{(M-2)!}{(M-1)^{k-2}}\left\{ {k-2 \atop M-2} \right\}\right] + \left(\frac{M-1}{M}\right)\left[\frac{(M-1)!}{M^{k-2}}\left\{ {k-2 \atop M-1} \right\}\right] \\
&= (1-\frac{1}{M})^{k-1}P_{M-1}(k-1) + \frac{M-1}{M}P_M(k-1)
\end{aligned}$$

for $k \geq 2$. Since we have a linear recursion, it's feasible to calculate $P_M(k)$ for any pair M, k.

2.4.1.3 Test Setup

Let $XS = XS(\tilde{\sigma})$ denote the saturation point of the integer sequence $\tilde{\sigma}$.

First we need to say if all the possible integers does not occur in the given sequence with sequence size k, then we say XS of this sequence is $k+1$.

- For a chosen M, we calculate $P_M(k)$ for every $k = 1, 2, 3, ...$
- Let $K = \{1, 2, \ldots\}$ is the set of all possible saturation points. Let S_m be a set such that $\{P_M(XS=k) : \forall k \in K\}$. Let P_{M_i} is an ordering on set S_m such that $P_{M,i-1} < P_{M,i}$ for all $i > 0$. Then cumulative histogram M_i is defined as

$$M_i = \sum_{j=0}^{i} P_{M,j}$$

Find the smallest i such that $M_{i+1} > \alpha$ Say $S_m(i) = \{P_{M,1}, P_{M,2}, ..., P_{M,i}\}$ and λ_1,λ_2 are the smallest and the largest elements of the set $\{k \,|k \in K \setminus S_m(i)\}$ respectively.

- Define another cumulative histogram X_i such that

$$X_j = \sum_{i=1}^{j} P_M(i)$$

and find the smallest j such that $X_{j+1} > \alpha$ and let us say that $k_{j+1} = \beta$.

- Lastly, find θ such that $0.99999 = \sum_{i=0}^{\theta} P(XS = i)$

One can see that $\lambda_1 < \beta < \lambda_2 < \theta$.

After applying this process for every M, next step is choosing proper b-value for a given n-bit sequence.

Choosing Proper b-value

For a chosen b, compute $\Lambda_1 = \lambda_1 b$, $\Lambda_2 = \lambda_2 b$, $B = \beta b$ and $\Theta = \theta b$. Then,

1. If $n > \Theta$ or $n < B$, then do not apply the test.
2. If $B \leq n \leq \Lambda_2$, then we use one-sided test such that:
   - If $\beta \leq XS(\tilde{\sigma})$, then do not eliminate sequence.
   - Otherwise eliminate sequence.
3. If $\Lambda_2 \leq n \leq \Theta$, then we use two-sided test such that:
   - If $\lambda_1 \leq XS(\tilde{\sigma}) \leq \lambda_2$, then do not eliminate sequence.
   - Otherwise eliminate sequence.

Here is the table of threshold points and proper b-values for some n-bit sequences:

b	M	λ_1	β	λ_2	θ
3	8	8	9	51	102
4	16	22	26	116	221
5	32	59	69	258	471
6	64	155	176	564	993
7	128	390	435	1222	2081

b	M	Λ_1	B	Λ_2	Θ
3	8	24	27	153	306
4	16	88	104	464	884
5	32	295	345	1290	2355
6	64	930	1056	3384	5958
7	128	2730	3045	8554	14567

α=0.01

b	M	λ_1	β	λ_2	θ
3	8	9	10	46	102
4	16	23	28	105	221
5	32	62	73	236	471
6	64	162	186	521	993
7	128	405	454	1135	2081

b	M	Λ_1	B	Λ_2	Θ
3	8	27	30	138	306
4	16	92	112	420	884
5	32	310	365	1180	2355
6	64	972	1116	3126	5958
7	128	2835	3178	7945	14567

α=0.02

b	M	λ_1	β	λ_2	θ
3	8	9	11	41	102
4	16	25	30	95	221
5	32	66	78	215	471
6	64	170	197	477	993
7	128	423	478	1049	2081

b	M	Λ_1	B	Λ_2	Θ
3	8	27	33	123	306
4	16	100	120	380	884
5	32	330	390	1075	2355
6	64	1020	1182	2862	5958
7	128	2961	3346	7343	14567

α=0.04

Table 2.13: Threshold Values for Saturation Point Test

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3	■	■	■	■								
4		■	■	■	■	■	■					
5					■	■	■	■	■	■		
6									■	■	■	■
7												

α=0.01

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3	■	■	■	■								
4		■	■	■	■	■	■					
5					■	■	■	■	■	■		
6									■	■	■	■
7												

α=0.02

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3	■	■	■	■								
4		■	■	■	■	■	■					
5						■	■	■	■	■		
6									■	■	■	■
7												

α=0.04

Table 2.14: Proper b values for Saturation Point Test for some $n-bit$ Sequences

Here are some test results:

α	b	M	l	worst $\alpha - set$	actual α
0.01	6	64	682	$(1, 154) \cup (565, 682)$	$0,009935328$
0.01	7	128	585	$(1, 434)$	$0,009947208$
0.02	6	64	682	$(1, 161) \cup (522, 682)$	$0,019817559$
0.02	7	128	585	$(1, 453)$	$0,019562961$
0.04	6	64	682	$(1, 169) \cup (478, 682)$	$0,039947476$
0.04	7	128	585	$(1, 477)$	$0,039753757$

Table 2.15: Saturation Point Test for n=4096

α	b	M	l	worst $\alpha - set$	actual α
0.01	3	8	42	$(1, 8)$	$0,002403259$
0.01	4	16	32	$(1, 25)$	$0,008692533$
0.02	3	8	42	$(1, 9)$	$0,002403259$
0.02	4	16	32	$(1, 27)$	$0,018959223$
0.04	3	8	42	$(1, 8) \cup \{42\}$	$0,035721276$
0.04	4	16	32	$(1, 29)$	$0,035433872$

Table 2.16: Saturation Point Test for n=128

2.4.2 Repeating Point Test

Repeating Point Test derives the first index of the repetition in the sequence. It is denoted by XR. First the sequence is converted to $b - bit$ integer sequence and then, starting from the first term, each term is compared to the predecessor terms.

Assume the first repetition appears at k^{th} point. That is, the first $k-1$ terms are distinct and the k^{th} term is equal to one of the first $k-1$ terms. Then, one can choose $k-1$ distinct integers out of M for the first $k-1$ terms and these terms can be managed in $(k-1)!$ ways. For the k^{th} element there are $k-1$ possible values. Therefore, there are

$$\binom{M}{k-1}(k-1)!(k-1)$$

and the probability of a repetition to occur at k^{th} index is

$$M^{-k}\binom{M}{k-1}(k-1)!(k-1).$$

2.4.2.1 Probability Distribution Function

The subject of Repeating Point Test is the index of integer, denoted by XR, where the first time we see repeating integer.

$$P(XR = k) = M^{-k}\binom{M}{k-1}(k-1)!(k-1) \tag{2.20}$$

2.4.2.2 Recursion

- $P_M(1) = 0$ for every $M = 1, 2, ...$
- $P_1(k) = 0$ for every $k = 1, 3, 4, ...$ and $P_1(2) = 1$
- $P_M(2) = 1/M$
- for $k \geq 3$ and $M \geq 2$;
  Since,

$$\begin{aligned}\binom{M}{k-1} &= \frac{M!}{(k-1)!(M-k+1)!} \\ &= \frac{M!}{(M-k+2)!(k-2)!}\frac{M-k+2}{k-1} \\ &= \binom{M}{k-2}\frac{M-k+2}{k-1}\end{aligned} \tag{2.21}$$

  we can write $P_M(k)$ as:

$$\begin{aligned}P_M(k) &= \frac{(k-1)!(k-1)}{M^k}\binom{M}{k-1} \\ &= \frac{(k-1)(k-2)!(k-1)}{M^k}\binom{M}{k-2}\frac{M-k+2}{k-1} \\ &= \left[\frac{(k-2)!(k-2)}{M^{k-1}}\binom{M}{k-2}\right]\frac{(k-1)(M-k+2)}{M(k-2)} \\ &= P_M(k-1)\frac{(k-1)(M-k+2)}{M(k-2)}.\end{aligned}$$

2.4.2.3 Test Setup

Let $XR = XR(\tilde{\sigma})$ denote the repeating point of the integer sequence $\tilde{\sigma}$.

- For a chosen M, we calculate $P_M(k)$ for every $k = 1, 2, 3, ...$

- Let $K = \{1, 2, \ldots\}$ is the set of all possible repeating points. Let S_m be a set such that $\{P_M(XR = k) : \forall k \in K\}$. Let P_{M_i} is an ordering on set S_m such that $P_{M,i-1} < P_{M,i}$ for all $i > 0$. Then cumulative histogram M_i is defined as

$$M_i = \sum_{j=0}^{i} P_{M,j}$$

  Find the smallest i such that $M_{i+1} > \alpha$ Say $S_m(i) = \{P_{M,1}, P_{M,2}, ..., P_{M,i}\}$ and λ_1, λ_2 are the smallest and the largest elements of the set $\{k|k \in K \setminus S_m(i)\}$ respectively.

- Find θ such that $0.99999 = \sum_{i=0}^{\theta} P(XR = i)$.

After applying this process for every b, next step is choosing proper b-value for a given n-bit sequence.

Choosing Proper b-value

For a chosen b, compute $\Lambda_1 = \lambda_1 b$, $\Lambda_2 = \lambda_2 b$, and $\Theta = \theta b$.

If Θ less than following Λ_2, than assign following Λ_2 to Θ'; otherwise assign Θ to Θ'. Then,

1. If $n > \Theta'$ or $n < \Lambda_2$, then do not apply the test.
2. If $\Lambda_2 \leq n \leq \Theta'$, then we use two-sided test such that:
   - If $\lambda_1 \leq XR(\tilde{\sigma}) \leq \lambda_2$, then do not eliminate sequence.
   - Otherwise eliminate sequence.

Here is the table of threshold points and proper b-values for some n-bit sequences:

b	M	λ_1	λ_2	θ
3	8	2	8	9
4	16	2	11	16
5	32	2	17	24
6	64	2	24	35
7	128	2	34	51
8	256	2	48	74
9	512	2	68	105
10	1024	2	97	150
11	2048	3	138	214
12	4096	3	194	304
13	8192	4	275	431

b	M	Λ_1	Λ_2	Θ	Θ'
3	8	6	24	27	44
4	16	8	44	64	85
5	32	10	85	120	144
6	64	12	144	210	238
7	128	14	238	357	384
8	256	16	384	592	612
9	512	18	612	945	970
10	1024	20	970	1500	1518
11	2048	33	1518	2354	2354
12	4096	36	2328	3648	3648
13	8192	52	3575	5603	5603

α=0.01

b	M	λ_1	λ_2	θ
3	8	2	7	9
4	16	2	11	16
5	32	2	15	24
6	64	2	22	35
7	128	2	31	51
8	256	2	44	74
9	512	3	64	105
10	1024	3	90	150
11	2048	4	127	214
12	4096	5	180	304
13	8192	6	255	431

b	M	Λ_1	Λ_2	Θ	Θ'
3	8	6	21	27	44
4	16	8	44	64	75
5	32	10	75	120	132
6	64	12	132	210	217
7	128	14	217	357	357
8	256	16	352	592	592
9	512	27	576	945	945
10	1024	30	900	1500	1500
11	2048	44	1397	2354	2354
12	4096	60	2160	3648	3648
13	8192	78	3315	5603	5603

α=0.02

b	M	λ_1	λ_2	θ
3	8	2	7	9
4	16	2	10	16
5	32	2	14	24
6	64	2	20	35
7	128	3	29	51
8	256	3	41	74
9	512	4	59	105
10	1024	4	82	150
11	2048	6	117	214
12	4096	7	165	304
13	8192	10	234	431

b	M	Λ_1	Λ_2	Θ	Θ'
3	8	6	21	27	40
4	16	8	40	64	70
5	32	10	70	120	120
6	64	12	120	210	210
7	128	21	203	357	357
8	256	24	328	592	592
9	512	36	531	945	945
10	1024	40	820	1500	1500
11	2048	66	1287	2354	2354
12	4096	84	1980	3648	3648
13	8192	130	3042	5603	5603

α=0.04

Table 2.17: Threshold Values for Repeating Point Test

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												

α=0.01

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												

α=0.02

	64	128	192	256	384	512	768	1024	1536	2048	3072	4096
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												

α=0.04

Table 2.18: Proper b values for Repeating Point Test for some $n-bit$ Sequences

Here are some test results:

α	b	M	l	worst $\alpha - set$	actual α
0.01	13	8192	315	$(2,3) \cup (276,315)$	$0,009915881$
0.02	13	8192	315	$(2,5) \cup (256,315)$	$0,019630694$
0.04	13	8192	315	$(2,9) \cup (235,315)$	$0,039127403$

Table 2.19: Repeating Point Test for n=4096

α	b	M	l	worst $\alpha - set$	actual α
0.01	5	32	25	$(18,25)$	$0,005201427$
0.02	5	32	25	$(16,25)$	$0,019581843$
0.04	5	32	25	$(15,25)$	$0,034812165$

Table 2.20: Repeating Point Test for n=128

2.4.3 Coverage Test

Coverage test is a kind of an integer test. The coverage of the test is described as the number of distinct elements in the sequence. The bit sequence should be converted into an integer sequence as every integer tests.

2.4.3.1 Probability Distribution Function

The subject of Coverage Test is the index of integer, denoted by XC.

$$P(XC = k) = \frac{k!}{M^l} \binom{M}{k} \left\{ {l \atop k} \right\}$$

2.4.3.2 Recursion

As stated in [16];

$$\begin{aligned}
P_l(k) &= \binom{M}{k} \left\{ {l \atop k} \right\} \frac{k!}{M^l} \\
&= \binom{M}{k} \frac{k!}{M^l} \left[\left\{ {l-1 \atop k-1} \right\} + k \left\{ {l-1 \atop k} \right\} \right] \\
&= \binom{M}{k} \frac{k!}{M^l} \left\{ {l-1 \atop k-1} \right\} + \binom{M}{k} \frac{k!}{M^l} k \left\{ {l-1 \atop k} \right\} \\
&= \frac{M-k+1}{m} \left[\binom{M}{k-1} \frac{(k-1)!}{M^{l-1}} \left\{ {l-1 \atop k-1} \right\} \right] + \frac{k}{M} \left[\binom{M}{k} \left\{ {l-1 \atop k} \right\} \frac{k!}{M^{l-1}} \right] \\
&= \left(1 - \frac{k-1}{M} \right) P_{l-1}(k-1) + \frac{k}{M} P_{l-1}(k)
\end{aligned}$$

2.4.3.3 Test Setup

This test differs from the other b-bit integer tests in testing. Before we get probability distribution, we need to know the integer sequence size.

Given sequence of length n, we need to apply our main method for every possible b values. We do not have any method for choosing proper b-value, previous b-bit integer tests can shed light on choosing it.

Here are some test results:

α	b	M	l	worst $\alpha - set$	actual α
0.01	12	4096	341	$(1, 318) \cup (337, 341)$	$0,009675332$
0.01	11	2048	372	$(1, 327) \cup (354, 372)$	$0,009084112$
0.01	10	1024	409	$(1, 320) \cup (355, 409)$	$0,008860071$
0.01	9	512	455	$(1, 284) \cup (321, 455)$	$0,008795681$
0.01	8	256	512	$(1, 209) \cup (234, 512)$	$0,007904103$
0.01	7	128	585	$(1, 123)$	$0,008068765$
0.02	12	4096	341	$(1, 319) \cup (337, 341)$	$0,014805103$
0.02	11	2048	372	$(1, 328) \cup (353, 372)$	$0,016213723$
0.02	10	1024	409	$(1, 322) \cup (354, 409)$	$0,01694815$
0.02	9	512	455	$(1, 286) \cup (319, 455)$	$0,019875364$
0.02	8	256	512	$(1, 210) \cup (233, 512)$	$0,014907138$
0.02	7	128	585	$(1, 123)$	$0,008068765$
0.04	12	4096	341	$(1, 320) \cup (336, 341)$	$0,03143676$
0.04	11	2048	372	$(1, 330) \cup (352, 372)$	$0,03525057$
0.04	10	1024	409	$(1, 324) \cup (352, 409)$	$0,037593169$
0.04	9	512	455	$(1, 287) \cup (317, 455)$	$0,034442849$
0.04	8	256	512	$(1, 211) \cup (231, 512)$	$0,036325143$
0.04	7	128	585	$(1, 124)$	$0,037431399$

Table 2.21: Coverage Test for n=4096

α	b	M	l	worst $\alpha - set$	actual α
0.01	11	2048	11	$(1, 9)$	$0,000271936$
0.01	10	1024	12	$(1, 10)$	$0,001576062$
0.01	9	512	14	$(1, 11)$	$0,000451931$
0.01	8	256	16	$(1, 13)$	$0,007757892$
0.01	7	128	18	$(1, 13)$	$0,002128124$
0.01	6	64	21	$(1, 14)$	$0,008616901$
0.01	5	32	25	$(1, 12) \cup (22, 25)$	$0,007214554$
0.01	4	16	32	$(1, 10)$	$0,001721586$
0.02	12	4096	10	$(1, 9)$	$0,010934609$
0.02	11	2048	11	$(1, 9)$	$0,000271936$
0.02	10	1024	12	$(1, 10)$	$0,001576062$
0.02	9	512	14	$(1, 12)$	$0,011731591$
0.02	8	256	16	$(1, 13)$	$0,007757892$
0.02	7	128	18	$(1, 14)$	$0,017113636$
0.02	6	64	21	$(1, 14)$	$0,008616901$
0.02	5	32	25	$(1, 13) \cup (22, 25)$	$0,013036612$
0.02	4	16	32	$(1, 11)$	$0,016965046$
0.04	12	4096	10	$(1, 9)$	$0,010934609$
0.04	11	2048	11	$(1, 10)$	$0,02654286$
0.04	10	1024	12	$(1, 10)$	$0,001576062$
0.04	9	512	14	$(1, 12)$	$0,011731591$
0.04	8	256	16	$(1, 13)$	$0,007757892$
0.04	7	128	18	$(1, 14)$	$0,017113636$
0.04	6	64	21	$(1, 14) \cup 21$	$0,033305177$
0.04	5	32	25	$(1, 14) \cup (22, 25)$	$0,03811816$
0.04	4	16	32	$(1, 11)$	$0,016965046$

Table 2.22: Coverage Test for n=128

CHAPTER 3

Conclusion

In this thesis, we have studied binary and integer tests. We propose a method to test single sequence using the exact distributions which are obtained already using recursive methods. Without these recursive relations, it takes exponential complexity to find probability distribution.

The contribution of the thesis and future works can be stated as follows:

- Using exact probability distributions, we provided the way to find applicable and meaningful boundaries for each tests. With these boundaries, our tests are applicable for singular sequences to be tested.
- We present test suite for single sequence and we gave the boundaries for the sequences of length 4096 and length 128 through the binary tests that are Weight Test, Number of Total Runs Test, Number of Runs of length 1 Test, Number of Runs of length 2 Test, Excursion Test of y=0 line, Excursion Test of y=1 line and b-bit integer tests which are Saturation Point test, Repeating Point Test, Coverage Test with suitable b values. We faced some troubles while determining suitable b values for b-bit integer tests. Nevertheless, we propose a new method. We choose three distinct values of significance level as $\alpha = 0.01$, $\alpha = 0.02$ and $\alpha = 0.04$.
- We conducted experiments using Microsoft Excel and JavaScript. We implemented all tests and made experiment on random sequences. As it is seen in 3.1 and 3.2, we apply our tests in some order to 100.000 random sequences of length 4096. While all test results(elimination ratio) are close to actual α, results of Number of Runs of length 1 Test and Excursion Test of y=1 line differ greatly. Therefore, we change the order as it is seen in 3.3 and 3.4 and we still saw the same difference Number of Total Runs Test and Excursion Test of y=1 line.
- For the future work, correlations between Number of Total Runs Test and Number of Runs of length 1 Test and correlations between Excursion Test of y=1 and Excursion Test of y=2 can be examined.

Experiment 1

Test	Random Set Size	Survivor Set Size	Elimination Ratio	Actual Alpha	Alpha
Weight	100000	98990	0.0101	0.0099253	0.01
All Runs	98990	98039	0.0096	0.0099250	0.01
Runs(a=1)	98039	97547	0.0050	0.0097146	0.01
Runs(a=2)	97547	96694	0.0087	0.0093352	0.01
Excursion(y=0)	96694	95761	0.0096	0.0097995	0.01
Excursion(y=1)	95761	95464	0.0031	0.0097995	0.01
Saturation(b=6)	95464	94544	0.0096	0.0099353	0.01
Saturation(b=7)	94544	93601	0.0100	0.0099472	0.01
Repeating(b=13)	93601	92698	0.0096	0.0099159	0.01
Coverage(b=7)	92698	91966	0.0079	0.0083374	0.01
Coverage(b=8)	91966	91252	0.0078	0.0079041	0.01
Coverage(b=9)	91252	90462	0.0086	0.0086069	0.01
Coverage(b=10)	90462	89640	0.0091	0.0087798	0.01
Coverage(b=11)	89640	88802	0.0093	0.0090542	0.01
Coverage(b=12)	88802	87971	0.0094	0.0095619	0.01

Table 3.1: Experiment 1

Experiment 2

Test	Random Set Size	Survivor Set Size	Elimination Ratio	Actual Alpha	Alpha
Weight	100000	99029	0.0097	0.0099253	0.01
All Runs	99029	98045	0.0099	0.0099250	0.01
Runs(a=1)	98045	97576	0.0048	0.0097146	0.01
Runs(a=2)	97576	96717	0.0088	0.0093352	0.01
Excursion(y=0)	96717	95728	0.0102	0.0097995	0.01
Excursion(y=1)	95728	95441	0.0030	0.0097995	0.01
Saturation(b=6)	95441	94496	0.0099	0.0099353	0.01
Saturation(b=7)	94496	93521	0.0103	0.0099472	0.01
Repeating(b=13)	93521	92580	0.0101	0.0099159	0.01
Coverage(b=7)	92580	91784	0.0086	0.0083374	0.01
Coverage(b=8)	91784	91016	0.0084	0.0079041	0.01
Coverage(b=9)	91016	90286	0.0080	0.0086069	0.01
Coverage(b=10)	90286	89542	0.0082	0.0087798	0.01
Coverage(b=11)	89542	88730	0.0091	0.0090542	0.01
Coverage(b=12)	88730	87940	0.0089	0.0095619	0.01

Table 3.2: Experiment 2

Experiment 3

Test	Random Set Size	Survivor Set Size	Elimination Ratio	Actual Alpha	Alpha
Weight	100000	99043	0.0096	0.0099253	0.01
Runs(a=1)	99043	98049	0.0100	0.0097146	0.01
Runs(a=2)	98049	97197	0.0087	0.0093352	0.01
All Runs	97197	96746	0.0046	0.0099250	0.01
Excursion(y=1)	96746	95810	0.0097	0.0097995	0.01
Excursion(y=0)	95810	95487	0.0034	0.0097995	0.01
Saturation(b=6)	95487	94567	0.0096	0.0099353	0.01
Saturation(b=7)	94567	93591	0.0103	0.0099472	0.01
Repeating(b=13)	93591	92726	0.0092	0.0099159	0.01
Coverage(b=7)	92726	91903	0.0089	0.0083374	0.01
Coverage(b=8)	91903	91190	0.0078	0.0079041	0.01
Coverage(b=9)	91190	90391	0.0088	0.0086069	0.01
Coverage(b=10)	90391	89523	0.0096	0.0087798	0.01
Coverage(b=11)	89523	88698	0.0092	0.0090542	0.01
Coverage(b=12)	88698	87839	0.0097	0.0095619	0.01

Table 3.3: Experiment 3

Experiment 4

Test	Random Set Size	Survivor Set Size	Elimination Ratio	Actual Alpha	Alpha
Weight	100000	99083	0.0092	0.0099253	0.01
Runs(a=1)	99083	98124	0.0097	0.0097146	0.01
Runs(a=2)	98124	97207	0.0093	0.0093352	0.01
All Runs	97207	96714	0.0051	0.0099250	0.01
Excursion(y=1)	96714	95737	0.0101	0.0097995	0.01
Excursion(y=0)	95737	95411	0.0034	0.0097995	0.01
Saturation(b=6)	95411	94493	0.0096	0.0099353	0.01
Saturation(b=7)	94493	93537	0.0101	0.0099472	0.01
Repeating(b=13)	93537	92594	0.0101	0.0099159	0.01
Coverage(b=7)	92594	91836	0.0082	0.0083374	0.01
Coverage(b=8)	91836	91141	0.0076	0.0079041	0.01
Coverage(b=9)	91141	90366	0.0085	0.0086069	0.01
Coverage(b=10)	90366	89576	0.0087	0.0087798	0.01
Coverage(b=11)	89576	88791	0.0088	0.0090542	0.01
Coverage(b=12)	88791	87943	0.0096	0.0095619	0.01

Table 3.4: Experiment 4

REFERENCES

[1] L. E. Bassham, III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications, Technical report, Gaithersburg, MD, United States, 2010.

[2] A. Doganaksoy, C. Calık, F. Sulak, and M. S. Turan, New randomness tests using random walk, in *National Cryptology Symposium II*, 2006.

[3] A. Doğanaksoy, F. Sulak, M. Uğuz, and O. Koçak, Recursive expressions for random walk statistics and a randomness test for binary sequences, submitted.

[4] A. Doğanaksoy, F. Sulak, M. Uğuz, O. Şeker, and Z. Akcengiz, New statistical randomness tests based on length of runs, Mathematical Problems in Engineering, 2015, 2015.

[5] A. Doğanaksoy, F. Sulak, M. Uğuz, O. Şeker, and Z. Akcengiz, Mutual correlation of nist statistical randomness tests and comparison of their sensitivities on transformed sequences, Turkish Journal of Electrical Engineering & Computer Sciences, 25(2), pp. 655–665, 2017.

[6] S. Kim, K. Umeno, and A. Hasegawa, Corrections of the NIST statistical test suite for randomness, IACR Cryptology ePrint Archive, 2004, p. 18, 2004.

[7] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Addison-Wesley, Boston, third edition, 1997, ISBN 0201896842 9780201896848.

[8] O. Koçak, F. Sulak, A. Doğanaksoy, and M. Uğuz, Modifications of knuth randomness tests for integer and binary sequences, submitted.

[9] A. L. Rukhin, Testing randomness: A suite of statistical procedures, 45, 04 2000.

[10] U. M. Maurer, A universal statistical test for random bit generators, J. Cryptology, 5(2), pp. 89–105, 1992.

[11] F. Sulak, Statistical analysis of block ciphers and hash functions, Middle East Technical University, 2011.

[12] F. Sulak, A new statistical randomness test: Saturation point test, International Journal of Information Security Science, 2(3), pp. 81–85, 2013.

[13] F. Sulak, New statistical randomness tests: 4-bit template matching tests, Turkish Journal of Mathematics, 41(1), pp. 80–95, 2017.

[14] F. Sulak, M. Uğuz, O. Koçak, A. Doğanaksoy, and G. TUBITAK UEKAE, On the independence of statistical randomness tests included in the nist test suite 2, Turkish Journal of Electrical Engineering & Computer Sciences, pages, 2016.

[15] M. Uğuz, A. Doğanaksoy, F. Sulak, and O. Koçak, R-2 composition tests: A family of statistical randomness tests for a collection of binary sequences, submitted.

[16] M. Uğuz, O. Koçak, F. Sulak, and A. Doğanaksoy, Css-5: Five randomness tests for collections of short sequences, submitted.