Journal of Modern Technology and Engineering
Vol.7, No.1, 2022, pp.74-84

ON A NEW NUMBER SYSTEM AND ITS APPLICATIONS

Nargiz Khankishiyeva*

Ege University, Izmir, Turkey

Abstract. In this paper, firstly, different number systems are considered, the efficiency, advantages and disadvantages of these number systems in the digital environment are discussed. Then a new number system is introduced, algorithms for arithmetic operations on this number system are developed. In the final section of the study some applications of the new number system are given.

Keywords: Number systems, Nuriyev Number System, cryptology, data compression, arithmetic operations on NNS.
Corresponding author: Nargiz Khankishiyeva, Ege University, Izmir, Turkey,
e-mail: *nargizkhankishiyeva@gmail.com*
Received: 21 January 2022; Revised: 12 March 2022; Accepted: 27 March 2022; Published: 29 April 2022.

1 Introduction

Information, images, sounds etc. in the memory of computers are shown as combinations of zeros and ones. The known number system containing 0 and 1 is the *binary number system*. This number system is commonly used by computers in terms of its easy applicability in the digital environment. However, one disadvantage of binary number system is that as it has two digits, even small numbers are represented by too many symbols. With the development of the technologies, there has been a great interest in number systems, even the computers using the *balanced triple number system* have been developed (Brusentsov, 1970). In this paper, firstly, the new number system (NURİYEV NUMBER SYSTEM- NNS) which consists of only 0 and 1 like binary number system is considered, then the algorithms of arithmetic operations on this number system are introduced, the data compression and cryptology applications are given.

2 Number Systems

The growth of numbers from the past to the present, depending on how people use them, has led to development of different number systems for the purpose of representing these large numbers. For example, only numbers up to 9999 could be expressed by the numerals of Cistercian number system, which was once used in many countries from England to Italy, from Spain to Sweden. In order to express larger numbers, this number system has been replaced by new ones.

Number systems are divided into two sections:

1. Non-positional number systems

2. Positional number systems

2.1 Non-Positional Number Systems

In these number systems, the value of symbols doesn't depend on their positions: Ancient Egypt number system, Roman number system and Cistercian number system etc.

For example, in Roman number system number 73 is written as LXXIII:

$$\underbrace{\mathbf{L}}_{\mathbf{50}}\underbrace{\mathbf{X}}_{\mathbf{10}}\underbrace{\mathbf{X}}_{\mathbf{10}}\underbrace{\mathbf{I}}_{\mathbf{1}}\underbrace{\mathbf{I}}_{\mathbf{1}}\underbrace{\mathbf{I}}_{\mathbf{1}}$$

Here, X is in the right side of L, it means 50+10+10=70, I is in the right side of X and it means $70+1+1+1=73$. In this number system, the value of symbols doesn't depend on their position. In this example, I appears three times and the value of I is the same at these three times and equals 1 and X appears two times and the value of X is the same at both times and equals 10.

Ancient Egypt number system: In this number system, digits are corresponding symbols of numbers $1, 10, 10^2,\ 10^3,\ 10^4,\ 10^5,\ 10^6,\ 10^7$ in decimal number system. The same symbol can be used in a number up to 9 times.

Roman number system: In this number system, instead of $1, 5, 10, 50, 100$ and 1000, they used the letters I, V, X, L, C, D, M from Latin alphabet as symbols. For example, number 96 is written as XCVI:

$$\underbrace{\mathbf{X}}_{\mathbf{10}}\underbrace{\mathbf{C}}_{\mathbf{100}}\underbrace{\mathbf{V}}_{\mathbf{5}}\underbrace{\mathbf{I}}_{\mathbf{1}}$$

Here, X is in the left side of C, it means $100 - 10 = 90$, I is in the right side of V and V is in the right side of C, therefore it means $90 + 5 + 1 = 96$.

***Cistercian number system*:** It is thought that this number system, which is known to belong to the 13^{th} century, was developed by the Cistercian priests, a branch of the Catholics and was widely used among priests.

2.2 Positional Number Systems

In this number systems, the value of symbols depends on their positions: Babylonian number system, Decimal number system, Binary number system and so on.

For example, in decimal number system the digit 2 appears three times in the number 222 and the value of '2' in first position is $2 \cdot 10^2 = 200$, the next value of '2' in the second position is $2 \cdot 10^1 = 20$ and the last value of '2' in the last position is $2 \cdot 10^0 = 2$. It means 2 has three different values according to three positions. Number systems are designed by this way are positional number systems.

Any number N in any p-based number system can be represented by coefficients and places:

$$N = a_k p^k + a_{k-1} p^{k-1} + \cdots + a_1 p^1 + a_0 p^0.$$

This number can be represented as follows:

$$N = \left(a_k a_{k-1} ... a_1 a_0\right)_p.$$

Positional number systems are developed independently of each other in Babylon, Maya tribe and India.

***Babylonian number system*:** The Babylonian Sexagesimal number system is one of the first developed positional number systems. Dividing an hour into 60 minutes, a minute into 60 seconds is the trace of sexagesimal number system.

Maya number system: This number system, belonging to the Maya tribe living in the Yucatan peninsula, is based on 20, and has the number zero.

Decimal number system: In this number system, the numbers consist of the elements of the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. In this number system, a number is represented as follows:

$$(a_k a_{k-1} ... a_1 a_0)_{10} = a_k 10^k + a_{k-1} 10^{k-1} + \cdots + a_1 10^1 + a_0 10^0.$$

The decimal number system didn't establish its throne in our lives immediately, it has taken the highest place among other number systems used by different peoples in different periods throughout history.

Binary number system: Although binary number system which is one of the old number systems with a minimum number of digits, was not widely used in the past, it is the most efficient number system for computers today.

The general form of the numbers in this number system is as follows:

$$(a_k a_{k-1} ... a_1 a_0)_2 = a_k 2^k + a_{k-1} 2^{k-1} + a_1 2^1 + a_0 2^0.$$

$p - q$ number system

If $q = p^m$, $m \epsilon N$, the number systems holding these bases are called $p - q$ number systems. As example for $p-q$ number system we can say octal and hexadecimal number systems.

Base8 (Octal): Octal number system is one of the number systems used in computers with the digits {0, 1, 2, 3, 4, 5, 6, 7}. Let's write 30_{10} in the octal number system:

$$36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 24 + 6 = 30_{10}.$$

Base16 (Hexadecimal): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} symbols are the alphabet of this number system. Here A, B, C, D, E, F represent respectively 10, 11, 12, 13, 14, 15 numbers in decimal number system. Let's write 30_{10} in the hexadecimal number system:

$$1E_{16} = 1 \cdot 16^1 + 14 \cdot 16^0 = 16 + 14 = 30_{10}.$$

Base32: Base32 is the numeral system which uses 32 as a base. This system uses 22 uppercase letters of alphabet and 10 digits as a symbol. However, there are many variations of this symbols and one of them offered by Douglas Crockford. In base32 offered by Douglas symbols are uppercase letters of alphabet without {I, O, L, U} and {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} the set of digits.

Base64: Base64 is not intended to represent numbers, but as an encryption method for transmitting bits in messages with text content. In base64, digits of decimal number system, lowercase and uppercase letters of alphabet and {+, \} are used for the symbol.

Kaktovik number system: The Kaktovik number has a history of about 25 years and was developed by the school teacher linguist William Bartley and his students in 1994. The symbols used by this number system and its equivalent in decimal number system are shown in the picture below

Figure 1: The digits of Kaktovik number system

We can divide positional number systems into two parts as traditional and non-traditional number systems

1. *Traditional number systems*: In the traditional number system, the value of the digits in the p-based number system mentioned above is non-negative integer powers (geometric progress) of the natural number p.

Table 1: Numbers in balanced ternary and decimal number system

Balanced ternary	Decimal
$\bar{1}101\bar{1}_3$	-52_{10}
$1\bar{1}0\bar{1}1_3$	52_{10}
$\bar{1}10_3$	-6_{10}
$1\bar{1}0_3$	6_{10}

2. *Non-traditional number systems*: Unlike traditional number systems, their base is unusual. Among these number systems, there are also mixed-based number systems.

Balanced number systems: As example for this number system, we can show the ternary number system. The digits of this number system are $\{\bar{1}, 0, 1\}$. Here $\bar{1}$ means (-1).

For example,

$$\bar{1}101\bar{1}_3 = (-1) \cdot 3^4 + 1 \cdot 3^3 + 0 \cdot 3^2 + 1 \cdot 3^1 + (-1) \cdot 3^0 = -52_{10}$$

$$1\bar{1}0_3 = 1 \cdot 3^2 + (-1) \cdot 3^1 + 0 \cdot 3^0 = 6_{10}$$

As it can be understood from these examples, in the balanced ternary number system, if the number starts with 1, it becomes positive, if the number starts with $\bar{1}$, it becomes negative number. It can be seen from examples, there is no need to write the sign (-) in front of the number in this number system.

$$1\bar{1}0\bar{1}1_3 = 1 \cdot 3^4 + (-1) \cdot 3^3 + 0 \cdot 3^2 + (-1) \cdot 3^1 + 1 \cdot 3^0 = 52_{10}$$

$$\bar{1}10_3 = (-1) \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^0 = -6_{10}$$

When we examine the table above, we see that negative and positive numbers are symmetrical in this number system, and for this reason the other name of this system is symmetric ternary number system.

Nega - position number systems: In these number systems, the base (q) is no longer a natural number, but a negative integer. For example, if $q = -2$, then the number system is called negabinary, if $q = -4$, then the number system is called negaquaternary and so on.

The general form of the number in this number system is as follows:

$$(a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-n} \dots)_{-q} =$$

$$= a_n(-q)^n + a_{n-1}(-q)^{n-1} + \dots + a_1(-q)^1 + a_0(-q)^0 + a_{-1}(-q)^{-1} + \dots a_{-n}(-q)^{-n} + \dots$$

Negabinary: It is a negative based number system with the digits {0,1}.

For example,

$$1101_{-2} = 1 \cdot (-2)^3 + 1 \cdot (-2)^2 + 0 \cdot (-2)^1 + 1 \cdot (-2)^0 = -3_{10}.$$

Negaquaternary: The digits of this number system are the elements of the set {0, 1, 2, 3}.

The main reason why these number systems are attractive to computers is that the negative sign ('-') in front of the negative numbers is never seen.

Complex-based number systems: In these number systems bases (q) are the complex numbers. For example, q=$2i$. In $2i$-based number system the digits are {0, 1, 2, 3}.

For example,

$$103_{2i} = 1 \cdot (2i)^2 + 0 \cdot (2i)^1 + 3 \cdot (2i)^0 = -1_{10}.$$

Mixed-based number systems:

Factorial number system: In this number system, numbers are generally represented as follows:

$$(\mathrm{a_n a_{n-1} \dots a_2 a_1}) = \mathrm{a_n \cdot n! + a_{n-1} \cdot (n-1)! + \dots a_2 \cdot 2! + a_1 \cdot 1!}$$

For example,

$$312_f = 3 \cdot 3! + 1 \cdot 2! + 2 \cdot 1! = 22_{10}.$$

Fibonacci number system: This number system is another example of a mixed-based number system, and the base of this number system consists of Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21, ...

For example, let's write the equivalent of 20 in the Fibonacci number system:

$$20_{10} = 13 + 5 + 2 = 101010_{Fi}.$$

Another example of mixed based number system is the number system whose base is a sequence of numbers 1, 2, 8, 48, ...

For example, in this number system the equivalent of 20 is written as follows:

$$20_{10} = 2 \cdot 8 + 2 \cdot 2 + 0 \cdot 1 = 220.$$

There is another type of positional number system called by *residue class-based number system.*

Residue class-based number system: To learn about this system, firstly, let's look through residue classes.

Let's take any number m from the set of natural numbers, for example, let $m = 5$ and call it a *module*. The remainders of the division of any natural number by $m = 5$ are $\{0,\ 1,\ 2,\ 3,\ 4\}$. By dividing the whole set of natural numbers into 5 classes add the numbers with the same remainder into the same class. In this example, residue classes are as follows:

$$a_n = 5n - 4,$$

$$a_n = 5n - 3,$$

$$a_n = 5n - 2,$$

$$a_n = 5n - 1,$$

$$a_n = 5\left(n - 1\right), \qquad n \in N^+.$$

All natural numbers A_{10} in this number system are represented as follows:

$$A_{10} = \left(a_1,\ a_2\ ,\ a_3,\ \ldots, a_k\right)_c,$$

Here,

$$a_i = A - \left\lfloor \frac{A}{p_i} \right\rfloor \cdot p_i,\ \ i = 1, 2, 3, .., k;$$

$\left\lfloor \frac{A}{p_i} \right\rfloor$ represents the whole part of division and p_i represents module and they are prime numbers.

2.3 The Efficient Number Systems For Computers

The number system used in the computer environment is the binary number system. So why do computers prefer the binary number system?

1. Its implementation requires technical mechanisms with two steady states (current-no current);

2. The presentation of information through only two situations takes place reliably and quickly;

3. It is the most suitable number system for the Boolean algebra mechanism, which is the basic logic of computers to perform logical information transformations;

4. Usage of the binary number system and arithmetic operations on this number system are very simple.

The disadvantage is that since it has two digits, it uses too many symbols to represent numbers compared the other number systems, numbers have more length in this number system. Computers use many number systems as well as binary number system. Octal and hexadecimal number systems are one of these number systems. While the binary number system is suitable for computers, it is not comfortable humans due to its length and unusual spelling. Computers automatically convert binary to decimal number system and vice versa for human convenience. However, for professional use of computers, it is necessary to learn the machine completely. For this, octal, hexadecimal, 32 and 64-based number systems are used.

The octal and hexadecimal number systems are almost as easy to use as the decimal number system. Compared to binary number system, three times (since $2^3 = 8$) fewer symbols are used in octal number system and four times (since $2^4 = 16$) fewer symbols are used in hexadecimal number system. In base32, each character represents, 20%-21% more bits than in the hexadecimal number system and 20%-21% fewer bits than in base64. Except those, *base36* and *base62* are also encryption bases used by computers. In base36, the digits of decimal number system $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and the uppercase letters of alphabet are used as symbols. In base62, the digits of decimal number system and the uppercase and lowercase letters of alphabet are used as symbols.

Apart from the binary number system, computers called 'SETUN' working with the balanced triple number system were produced at Moscow State University in 1962-1965 and the chief of this machine, Brusentsov, wrote that, the features of balanced triple number system were not used well enough in this machine. The biggest advantage of the symmetric triple number system is that the (-) sign is not in front of the number. Brusentsov wrote in his studies that subtraction is 1.59 times and multiplication is 2.5 times faster in the balanced number system than the binary number system. Except this, he wrote that, triple logic ('no'$\equiv -1$, 'maybe, undecided'$\equiv 0$, 'yes'$\equiv 1$) is more common in life and combination of binary control signals with triplet is possible.

The assignment of elements in the value of a symbol occurs with the help of signals. Recognition of the value represented by the signal depends on measuring value of a physical parameter (e. g. current strength, potential difference, field strength) that is the carrier of the signal. According to Brusentsova, when these controlled triple signals are sent in three different states over a signal cable, it demands two times fewer number of connections than when binary signals are sent on two different cables (Brusentsov, 1970).

When the efficiency on number systems in the digital environment is mentioned, besides ease of the arithmetic operations in these number systems, it is also meant that the number system is economical. The number system being economical means that it can express more numbers with a certain number of symbols.

Let's say we are looking for which x-based number system can represent with n symbols. Then, if there are $\frac{n}{x}$ steps and x digits in each, there can be written $x^{\frac{n}{x}}$ maximum numbers with n symbols. Let's look at this expression as a function dependent on the variable x and this variable can take all positive real numbers, let's find the value of the variable x for the maximum value of the function.

$$\frac{dy}{dx} = \frac{-n}{x^2} \cdot lnx + \frac{n}{x} \cdot x^{\frac{n}{x}-1} = nx^{\frac{n}{x}-2} \cdot (1 - lnx) = 0,$$

$$lnx = 1,\ x = e = 2,718281...$$

e is an irrational number and the nearest integer to this number is 3. For this reason, ternary number system is the most economical number system for computers (Fomin, 1987). Although ternary number system was theoretically very good for computers, computers named 'SETUN', which were made based on ternary number system couldn't reach the level of modern computers. Therefore, the binary number system still maintains its superiority in the digital environment.

The biggest reason why the binary number system can be easily integrated into the digital environment is that it consists of 0 and 1. As the new number system mentioned above (NNS), which was proposed in the study Nuriyev et al (2016), consists of 0 and 1, its applications in digital environment can be much more efficient for computers.

3 New Number System (Nuriyev Number System - NNS)

The number system N^k $(k = 1,\ 2,\ 3,\ ...)$ proposed below is non-positional number system as a base. There is no zero value in this system. Binary numbers with different lengths can be used in a base, here k is the base. For $k = 1$, in the number system N^1 , here exists only one value with one digit. Here, '0' means '1+'.

$$(1)_N^1 = 1_{10},$$
$$(0)_N^1 = (1+)_{10},$$
$$(01)_N^1 = 1 + 1 = 2_{10},$$
$$(00)_N^1 = 1 + 1+ = (2+)_{10},$$
$$(001)_N^1 = 1 + 1 + 1 = 2 + 1 = 3_{10}, ...$$

By this method we can write equivalents of all numbers in N^1 number system forever.

For $k = 2$, we can write all numbers in N^2 number system as follows:

$$(01)_N^2 = 1_{10},$$
$$(10)_N^2 = 2_{10},$$
$$(11)_N^2 = 3_{10},$$
$$(00)_N^2 = (3+)_{10},$$
$$(0001)_N^2 = 4_{10},$$
$$(0010)_N^2 = 5_{10},$$
$$(0011)_N^2 = 6_{10},$$
$$(0000)_N^2 = (6+)_{10}, ...$$

For $k = 3$, all numbers can be represented in N^3 number system as follows:

$$(001)_N^3 = 1_{10}$$
$$(010)_N^3 = 2_{10}$$
$$(011)_N^3 = 3_{10}$$
$$(100)_N^3 = 4_{10}$$
$$(101)_N^3 = 5_{10}$$
$$(110)_N^3 = 6_{10}$$
$$(111)_N^3 = 7_{10}$$
$$(000)_N^3 = (7+)_{10}$$

$$(000001)_N^3 = (7+1)_{10} = 8_{10}$$

$$(000010)_N^3 = 9_{10}, ...$$

By this method all numbers can be represented in N^k system forever. Thus, we can give the following general formula for all numbers in N^k number system (Andreeva & Falina, 1999).

$$\left(\underbrace{\underbrace{0\ldots0}_{k}\underbrace{0\ldots0}_{k}\ldots\underbrace{a_{k-1}\ldots a_0}_{k}}_{m}\right)_N^k = \left(\left(\frac{m}{k}-1\right)\cdot\left(2^k-1\right)+\left(a_{k-1}\cdot 2^{k-1}+\ldots a_1\cdot 2^1+a_0\right)\right).$$

4 Arithmetic Operations On NNS

4.1 Addition Operation

It is very important to write algorithms of arithmetic operations on NNS in order to learn these operations deeply. In following, there are given algorithm of addition operation on NNS:

Let's add two numbers with the length of $\boldsymbol{m}$ and $\boldsymbol{l}$ in N^k number system. The addition operation is done by dividing numbers into k digits:

$\underbrace{0\ldots0}_{k}\underbrace{0\ldots0}_{k}\ldots\underbrace{a_{k-1}\ldots a_0}_{k}$ is the number with the length of $\boldsymbol{l}$ and $\underbrace{0\ldots0}_{k}\underbrace{0\ldots0}_{k}\ldots\underbrace{b_{k-1}\ldots b_0}_{k}$ is the number with the length of $\boldsymbol{m}$.

a) First, let's look at the bits above and below in the first k digits and the carry number on them:

1. If both are 0, let's write 0 below and move on.
2. If both are 0 and there is a carry, or one is 0 and the other is 1 and no carry, then write 1, write 0 for carry and continue.
3. If both are 1 and there is no carry, or one is 1 and the other is 0 and there is one carry, then write 0 down and give one carry for the next place.
4. If both are 1, and the carry is one bit then write one below and add one carry to the next place.

b) For the (k+1)th digits let's look at the bits above and below and the carry number on them:

1. If both are 0 and have a carry, or one is 0 and the other is 1 and no carry, then write k bits of zeros ($\underbrace{00\ldots0}_{k}$) and add ($\underbrace{00\ldots0}_{k-1}1$) to the previous k digits and continue.
2. If one is 1 and the other is 0 and there is a carry, then add 2k of zeros ($\underbrace{00\ldots0}_{k}\underbrace{00\ldots0}_{k}$) below and add ($\underbrace{00\ldots0}_{k-1}1$) twice to the previous k digits and continue.

c) Let's look at the zero bits above and below in the next k digits:

1. If both are zero, let's add two zeros.
2. If one is zero and other is absent, let's add a zero.

4.2 Subtraction Operation

Let's say we need to subtract a number with the length of m from a number with the length of l. Subtraction is done by dividing the numbers into k digits.
a) In the first k digits let's look at the bits above and below and the borrow on them:

1. If both are 0 (both are 1) and there is no borrow or 1 bit above and 0 bit below and 1 borrow, write down 0 and give to the borrow 0 for the next place.
2. If bit above is 1 and bit below is 0 and there is no borrow, write down 1 and give borrow 0 for the next place.
3. If bit above is 0 and bit below is 1 and there is no borrow, if both are 0 (both are 1) and there is one borrow, write down 1 and give to the borrow 1 for the next place.
4. If bit above is 0 and bit below is 1 and one borrow, write down 0 and give to the borrow 1 for the next place.

b) In the next (k) digits: the above and below bits and borrow on them are checked:

1. If the bit above is 0, bit below and borrow is absent, write 0 to the result and continue.
2. If both bits are 0 and borrow is absent, don't write down anything and go on.
3. If both are zero and borrow is one, remove zeros above and below and remove k times of zeros $(\underbrace{00\ldots0}_{k})$ and remove $(\underbrace{00\ldots0}_{k-1}1)$ from difference in first (k) digits.
4. If above zero and no bit below and borrow is one, remove k times of zeros $(\underbrace{00\ldots0}_{k})$ above and remove $(\underbrace{00\ldots0}_{k-1}1)$ from difference in first (k) digits.

4.3 Multiplication Operation

Let's multiply two numbers of length l and m. Let's call the result product. Except the first k digits, the zeros of multiplicand and multiplier are replaced by 1 bit.

a) First look at the bits above and below.
b) Every bit of the number below is taken and multiplied by all bits of the number above.
c) The result of each multiplication is written by shifting one bit to the left.
d) Due to do addition operation the products is divided into k bits column.
e) Then starting from the left column, the addition process is done one under the other. Addition is done separately for each column.
f) If there is k of 1 $(\underbrace{11\ldots1}_{k})$in the columns those are replaced with k of zeros $(\underbrace{00\ldots0}_{k})$ and this k of zeros added to the product.
g) After the addition process is completed within the column, if there is any carry, the carry is added to the result in the same column and the addition process is continued until the carry is 0, and k of 0 $(\underbrace{00\ldots0}_{k})$ are added to the product each time as much as the carry ones.

h) If the carry is 0, the value found at the end of the column is added to the next column and k of 0 ($\underbrace{00\ldots0}_{k}$) is added to the product by the numbers of ones of the same value.

i) If the result of the operation in the column is k of 1 ($\underbrace{11...1}_{k}$), k of 0 ($\underbrace{00\ldots0}_{k}$) is added to the product.

j) The last result found in the last column is written to the multiplication as it is.

4.4 Division Operation

Division on the new number system is done with the following algorithms:

Suppose that a number with the length of l need to be divided by the number with the length of m.

$\underbrace{0\ldots0}_{k}\underbrace{0\ldots0}_{k}\ldots\underbrace{a_{k-1}\ldots a_0}_{k}$ is dividend and $\underbrace{0\ldots0}_{k}\underbrace{0\ldots0}_{k}\ldots\underbrace{b_{k-1}\ldots b_0}_{k}$ is divisor.

a) Except of the first k digit of the divisor and dividend number, zeros are replaced by 1.

b) Division starts from the left and progresses in each step by the length of the divisor.

c) Since the numbers consist of 0 and 1, the quotient is either 1 or it doesn't exist.

d) If there is a divisor in the first m digits, 1 is written to the quotient and if it is still greater than the divisor in the same m digits the division process is continued, and the quotients are added to each other.

e) If there is any remaining after the m-step operation is done, it is added to the last digit.

5 New Number System's Applications

5.1 Nuriyev Data Compression

Today, it is very important to compress and reduce the size of binary images, which have a wide range of applications in areas such as text, fingerprints and maps due to the large amount of storage. For the compression of binary images, coding methods enable the data forming binary images to be expressed with fewer bits have been developed. One of these methods is NDC (NURİYEV DATA COMPRESSİON) methods (Tanır & Nuriyeva, 2017). The basis of this method is based on coding the sequential numbers of repeated data by writing their equivalents in different bases of NNS (Nuriyev et al. 2016) Binary images consist of black and white colors. '1' symbolizes black color, '0' symbolizes white color. In NDC, in the first step, the number of repetitions of 0s and 1s representing white and black colors is written. In the second step, the repetition numbers are written in a such base of NNS that can express less bits. In this way, the coding process is completed. Detailed information about application of NNS in data compression is given in Tanır et al. (2016).

5.2 Cryptology

It is the science that encrypts and sends information according to a certain rule in order to ensure a secure flow of information between two institutions. One of these encryption methods was proposed by Nuriyeva and Karatay (2017). Encryption scheme: This scheme consists of repeatedly encoding the repeated bit numbers of information in different bases of NNS number system. For example, if encryption is done in the bases of N^3, N^2, N^4 respectively, the encryption/decryption method would be 324. This key will be sent to the other party securely with the RSA encryption scheme. Detailed information about this encryption method is given in Nuriyev et al. (2017).

6 Conclusion

In this paper, the number systems used in different times of history are mentioned, the number systems which used and still being used in the digital environment and their efficiency in the digital platform are discussed. Information about NNS is introduced, which is new among these number systems and arithmetic operations on NNS and their algorithms are given. The applications of NNS in the computer environment have been reviewed and it has been observed that it is efficient in compressing binary images by referring to previous studies.

References

Andreeva, E., Falina, I. (1999). *Computer Science: Number Systems and Computer Arithmetic.* Moscow, Laboratory of Basic Knowledge, 247 p (in Russian).

Brusentsov, N.P. (1970). *On the use of ternary code and three-valued logic in digital machines.* Computing and cybernetics. 7th ed., Moscow, The Publisher of Moscow University, 3-33 (in Russian).

Brusentsov, N.P., Maslov, S.P., Alvares, J.R., Zhogolev, E.A. (2015). Development of ternary computers of Moscow State University Russian Virtual Computer Museum. Retrieved January 19.

Fomin, S.V. (1987). *Number Systems. Series of "Popular Lectures in Mathematics".* (Vol 40), Moscow, Nauka, 48 p (in Russian).

Kasatkin, V.N. (1982). *New on Number Systems.* Kiev: Visha shkola. Golovnoe izdotelstvo, Library of the School of Physics and Mathematics. Maths. 96 p (in Russian).

Knuth, D.E. (1998). *The art of computer programming* (vol. 2). Seminumerical algorithms, 3rd ed., Boston, Addison - Wesley.

Kovrijenko, G.A. (1984). *Number Systems and Binary Arithmetic.* Kiev: Radyanska Shkola, 82 p (in Russian).

Nuriyeva, F., Karatay, M. (2017). On the analysis of an encryption scheme, *Theoretical and Applied Aspects of Program Systems Development*, 04-08 December, Kiev, Ukraine.

Nuriyev, U., Nuriyeva, F., Sadık T. (2016). On a New Number System. *Proceeding of the International Conference on Computer Science and Engineering (* pp. 825-827), *(UBMK 2016),* Namık Kemal University, Tekirdağ, Turkey, October 20-23.

Tanır, D., Nuriyeva, F. (2017). A New Method for Lossless Compression of Binary Images. *International Artificial Intelligence and Data Processing Symposium*, Malatya, Turkey, September 16-17, (IEEE Xplore Digital Library).

Ore, O. (2017). Invitation to number theory (Vol. 49). American Mathematical Soc.

Ağacı, E. Why is the Inyupik's Kaktovik Number System One of the Best Counting Methods? `https://evrimagaci.org/inyupiklerin-kakvotik-sayi-sistemi-neden-en-iyi-sayma-yontemlerinden-biridir-9961`

Wikipedia, (2022). Base32, https://en.wikipedia.org/wiki/Base32

Wikipedia,(2021). Base36, https://en.wikipedia.org/wiki/Base36

Wikipedia, (2022). Base62, https://en.wikipedia.org/wiki/Base62

Wikipedia, (2022). Base64, https://en.wikipedia.org/wiki/Base64