Highlights

Generating the $\mathbf{n^{th}}$ Lexicographical Element of a Mathematical *k*-Permutation using Permutational Number System

Deepesh Patel

- k-Permutation
- Factorial Number System
- Combinatorial Number System
- Permutational Number System

Generating the n$^{\text{th}}$ Lexicographical Element of a Mathematical k-Permutation using Permutational Number System

Deepesh Patel[a]

[a] *Sr. Manager R&D, Sincro Digital Marketing, Lunkad Sky One Corporate Park, 3rd Floor, New Airport Road, Viman Nagar, Pune, 411014, MH, India*

Abstract

This paper is about generating the n$^{\text{th}}$ lexicographical element of a mathematical k-permutation using a proposed concept called permutational number system and its mapping called Deep code to and from the n^{th} k-permutation. Permutational Number System or Permutadic for short is a number system based on permutations of numbers. The main application of the permutational number system is to rapidly compute k-permutation at a given position n (also known as n$^{\text{th}}$ k-permutation), in the lexicographical ordering starting from zero, without explicitly computing all the permutations preceding it. This concept is important because the total number of permutations can grow astronomically large. For instance, a number of permutations of 100 elements selected 50 at a time are ${}^{100}P_{50} = 3.068518756 * 10^{93}$, which is way beyond the practical limit to be generated sequentially to reach up to the desired permutation.

Keywords: Combinatorics, Number Theory, Factorial Number System, Factoradic, Combinatorial Number System, Combinadic, Permutational Number System, Permutadic

Email address: `patel.deepesh@gmail.com` (Deepesh Patel)

Preprint submitted to Journal of Combinatorial Theory, Series B *July 21, 2022*

1. Introduction

The permutational number system defined in this article is a generalization of the factorial number system[1]. In the field of combinatorics, the factorial number system, also called factorial base or factoradic, is a mixed radix number system used primarily for numbering unique permutations. From the definition of the factorial number system and Lehmer code, we know that by converting a number less than $n!$ to factorial number system representation, we can obtain a sequence of $\leq n$ digits that can be converted to a permutation of n using them as Lehmer code.

Similarly, a Permutational Number System is a numeral system adapted to numbering k-permutations. Any non-negative integer n can be represented in permutational number system which is a sequence of $k+1$ digits/numbers that can be converted to k-permutation of s unique elements using them as Deep code if $n < {}^{s}P_k$. Deep code is a generalization of Lehmer code that maps to and from k-permutations instead of unique-permutations. Deep code is explained in section (7).

Similar to the combinatorial number system[2] (also known as combinadic), permutational number system has a fixed number of digits/values but unlike combinadic, permutational number system is a mixed-radix number system. Ranking and un-ranking of k-permutation using permutadic and deep-code is depicted in the figure (1)

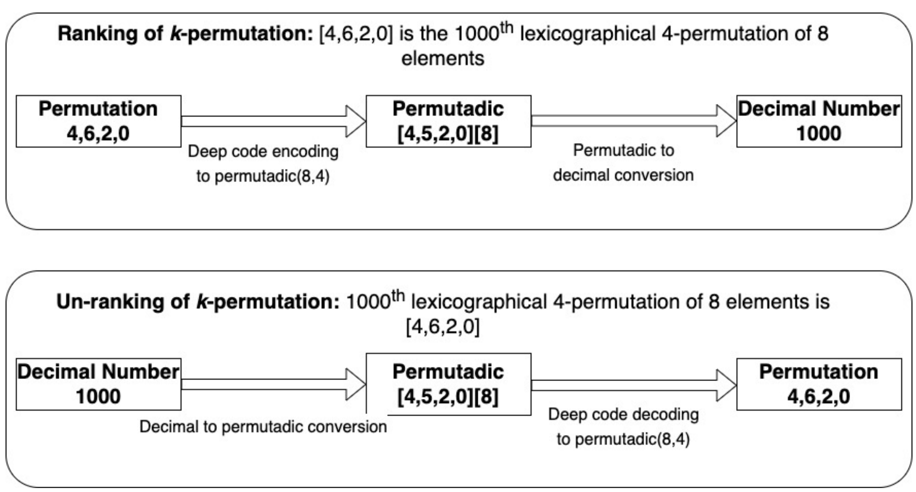

Figure 1: conversion of decimal number to/from k-permutation

2. Permutational Number System

2.1. Definition

Permutational number system of size s and degree k, for some positive integers s and k where $1 \leq k \leq s$, is a correspondence between natural numbers (starting from 0) and k-permutations, represented as a sequence $[C_{k-1}, C_{k-2}, C_{k-3} \dots C_1, C_0][s]$ where $C_i \in \mathbb{N}$.

Permutational number system of size s and degree k also referred to as permutadic(s, k) for short, is a mixed radix number system that has a unique representation for all natural numbers. The number n corresponding to the permutadic(s, k) string -

$$[C_{k-1}, C_{k-2}, C_{k-3} \dots C_1, C_0][s]$$

is expressed by the following equation -

$$n = \sum_{i=1}^{k} {}^{s-i}P_{k-i} \cdot C_{k-i} \tag{1}$$

where ${}^{s-i}P_{k-i}$ is the place value for the i^{th} digit/number from right, (and hence $C_{k-i} \leq {}^{s-i}P_{k-i} \forall i > 1$ and $C_{k-1} = n/{}^{s-1}P_{k-1}$) and the function ${}^{s}P_{k}$ represents the number of k-permutations from a set of size s and is defined as –

$${}^{s}P_{k} = \frac{s!}{(s-k)!} \tag{2}$$

2.2. Example:

Integer 5050 can be represented in permutadic$(8, 5)$ as [60022][8] because -

$5050 = {}^{7}P_{4}(6) + {}^{6}P_{3}(0) + {}^{5}P_{2}(0) + {}^{4}P_{1}(2) + {}^{3}P_{0}(2)$

2.3. First 40 permutadic$(5, 3)$ numbers(0 omitted)

Table (1) lists the decimal numbers and their permutadic$(5, 3)$ equivalent from 1 to 40. Place values of the digits are ${}^{4}P_{2}$, ${}^{3}P_{1}$ and ${}^{2}P_{0}$ from most significant value to least significant value, respectively.

Table 1: Permutadic$(5,3)$ representation of integers from 1 to 40

Decimal	Permutadic$(5,3)$	Decimal	Permutadic$(5,3)$
1	[0, 0, 1][5]	21	[1, 3, 0][5]
2	[0, 0, 2][5]	22	[1, 3, 1][5]
3	[0, 1, 0][5]	23	[1, 3, 2][5]
4	[0, 1, 1][5]	24	[2, 0, 0][5]
5	[0, 1, 2][5]	25	[2, 0, 1][5]
6	[0, 2, 0][5]	26	[2, 0, 2][5]
7	[0, 2, 1][5]	27	[2, 1, 0][5]
8	[0, 2, 2][5]	28	[2, 1, 1][5]
9	[0, 3, 0][5]	29	[2, 1, 2][5]
10	[0, 3, 1][5]	30	[2, 2, 0][5]
11	[0, 3, 2][5]	31	[2, 2, 1][5]
12	[1, 0, 0][5]	32	[2, 2, 2][5]
13	[1, 0, 1][5]	33	[2, 3, 0][5]
14	[1, 0, 2][5]	34	[2, 3, 1][5]
15	[1, 1, 0][5]	35	[2, 3, 2][5]
16	[1, 1, 1][5]	36	[3, 0, 0][5]
17	[1, 1, 2][5]	37	[3, 0, 1][5]
18	[1, 2, 0][5]	38	[3, 0, 2][5]
19	[1, 2, 1][5]	39	[3, 1, 0][5]
20	[1, 2, 2][5]	40	[3, 1, 1][5]

2.4. Comparison of factoradic, combinadic and permutadic

Table (2) depicts the comparison between factoradic, combinadic and permutadic.

3. Uniqueness of the permutadic representation

Any radix (either standard or mixed) base representation is unambiguous and complete, and every number can be represented in one and only one way because the sum of respective weights/place-value multiplied by the highest number allowed at index is always the next weight minus one. It might not be clear at first sight in the case of permutadic representation but can easily be proved with mathematical induction as explained in the following Theorem(1) -

Table 2: Comparison between number systems

Number system	Fixed width	Mixed radix	Primary Use
Combinatorial number system	Yes	No	Compute n^{th} combination
Factorial number system	No	Yes	Compute n^{th} unique permutation using Lehmer code
Permutational number system	Yes	Yes	Compute n^{th} k-permutation using Deep code

Theorem 1. *For all $s, k \in \mathbb{N} | k \leq s$,*

$$^{s}P_{k} - 1 = \sum_{i=1}^{k} {}^{s-i}P_{k-i} \cdot (s-i) \tag{3}$$

Proof of theorem 1. As a base case we can see that for $k = 1$, Equation(3) is true.

Assume the induction hypothesis that Equation (3) holds for some particular k.

By adding $({}^{s}P_{k} \cdot s)$ to both sides in Equation (3) we get,

$$^{s}P_{k} - 1 + ({}^{s}P_{k} \cdot s) = ({}^{s}P_{k} \cdot s) + \sum_{i=1}^{k} {}^{s-i}P_{k-i} \cdot (s-i)$$

$$\Rightarrow^{s} P_{k} \cdot (s+1) - 1 = \sum_{i=1}^{k+1} {}^{s+1-i}P_{k+1-i} \cdot (s+1-i) \tag{4}$$

Also, from Equation(2) -

$$\frac{^{s+1}P_{k+1}}{^{s}P_{k}} = \frac{(s+1)!}{(s+1-k-1)!} \cdot \frac{(s-k)!}{(s)!}$$

$$\Rightarrow \frac{^{s+1}P_{k+1}}{^{s}P_{k}} = (s+1)$$

$$\Rightarrow^{s+1} P_{k+1} =^{s} P_{k} \cdot (s+1) \tag{5}$$

from Equation(4) and Equation(5) -

$$^{s+1}P_{k+1} - 1 = \sum_{i=1}^{k+1} {}^{s+1-i}P_{k+1-i} \cdot (s+1-i)$$

which proves Theorem(1)

Also, from Equation(5), the highest value allowed for C_{k-i} is $(s-i)$ $\forall i | i \geq 2$ and $i \leq k$ because the highest value of the remainder is always 1 less than the divisor.

Which along with Theorem(1) satisfies the condition required for the uniqueness of representation in a mixed-radix number system. Hence we can say that there is a unique representation of all $n \in \mathbb{N}$ in permutational number system and we can interpret permutadic as a type of number system to represent all $n \in \mathbb{N}$.

4. Conversion from decimal to permutational number system:

To compute the permutadic of a natural number n, divide the number by ${}^{s-1}P_{k-1}$ and find the quotient q and remainder r. Quotient q is the most significant value of permutadic representation of n. To find the next significant digit/value, repeat the process by dividing r by ${}^{s-2}P_{k-2}$ and continue the process of storing the quotient and dividing the remainder by ${}^{s-i}P_{k-i}$, $\forall\ i \in A$, where $A = \{x : x \in \mathbb{N}, 1 \leq x \leq k\}$

4.1. Algorithm (Convert decimal to permutadic):

I/P :- s, k, n

for $i = 1$ to k, calculate,

- $C_{k-i} = n / {}^{s-i}P_{k-i}$
- $n = n \pmod{{}^{s-i}P_{k-i}}$

O/P :- $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0][s]$

4.2. Optimization of algorithm

Note that ${}^{s}P_k$ can be calculated from ${}^{s-1}P_{k-1}$ using the following relationship –

$$ {}^{s}P_k = s \cdot^{s-1} P_{s-k} \tag{6}$$

also, the least significant place value in permutadic representation is always

$$ {}^{s-k+1}P_0 \tag{7}$$

We can use Equation (6) and Equation (7) to optimize the above-mentioned algorithm such that we do not need to explicitly calculate the multiplier ${}^{s-i}P_{k-i}$ at each step. Steps for the optimized algorithm are as follows -

I/P :- s, k, n
$j = s - k + 1$
for $i = 0$ to $k - 2$

- $C_i = n \pmod j$
- $n = n/j$
- $j = j + 1$

$C_{k-1} = n$
O/P :- $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0][s]$

4.3. Example:

Table (3) depicts the conversion of 5050 to a permutational number system of size 8 and degree 5

5. Conversion from permutational number system to decimal

To convert the permutational number $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0][s]$ to decimal equivalent, the straightforward way is to multiply each C_i with ${}^{s-i}P_{k-i}$ and add the result, as expressed in Equation (1). But again we can apply the optimization from Equation (6) and Equation (7) so that we do not need to calculate the place-value explicitly. In the below mentioned algorithm, a variable *multiplier* is holding the place value.

Table 3: Conversion of decimal integer 5050 to Permutadic of size 8 and degree 5

n	Divisor	Quotient	Remainder
5050	$s-k+1=4$	$5050/4=1262$	2
1262	5	$1262/5=252$	2
252	6	$252/6=42$	0
42	7	$42/7=6$	0
6	NA	NA	$6(=n)$
Output $=[6,0,0,2,2][8]$			

5.1. Algorithm: Convert permutadic to decimal

I/P :- $[C_{k-1}, C_{k-2}, C_{k-3} \dots C_1, C_0][s]$

$sum = C_0$

$counter = s - k + 1$

$multiplier = counter$

for $i = 1$ to $k-1$, calculate,

- $sum = sum + (C_i * multiplier)$
- $counter = counter + 1$
- $multiplier = multiplier \cdot counter$

O/P :- sum

5.2. Example:

Table (4) depicts the conversion of permutational number $[6,0,0,2,2][8]$ to decimal equivalent

Table 4: Conversion of permutadic [6, 0, 0, 2, 2][8] to decimal

C_i	$counter$	$multiplier$	sum
2	NA	1	$2 \cdot 1 = 2$
2	$s-k+1=4$	$1 \cdot 4 = 4$	$2 + (2 \cdot 4) = 10$
0	5	$4 \cdot 5 = 20$	$10 + (0 \cdot 20) = 10$
0	6	$20 \cdot 6 = 120$	$10 + (0 \cdot 120) = 10$
6	7	$120 \cdot 7 = 840$	$10 + (6 \cdot 840) = 5050$
Output $=5050$			

6. Factorial number system as a special case of permutational number system

Theorem 2. *The permutadic representation of the natural number n contains the same sequence of numbers as in factoradic representation of n when $s = k$, $\forall n < s!$*

Proof of Theorem 2. for $s = k$, ${}^sP_k = \frac{s!}{s-s!} = s!$

from Equation (1), when $s = k$

$$n = \sum_{i=1}^{k} (s-i)! \cdot C_{k-i} \tag{8}$$

Which denotes the n^{th} integer in factorial number system when $n < s!$.

6.1. Example:

Say, $s = k = 5$ and $n = 100$ then, $s! = 5! = 120$ which satisfies constraint for permutational number system representation of n to be same as factorial number system representation.
And for,

$$(s, k, n) = (5, 5, 100)$$

permutational number system representation is:

$$[4, 0, 2, 0, 0][5]$$

and factoradic representation is:

$$[4, 0, 2, 0, 0]$$

Both of these representations are similar after omitting s from the permutadic representation. We also need to omit any leading zeros that we will get in case of larger values of s, for example for $s = k = 6$, permutadic representation of 100 is -

$$[0, 4, 0, 2, 0, 0][6]$$

7. Deep Code

7.1. Definition

Deep Code is a particular way to encode each possible k-permutation of s elements to a sequence of n numbers (0 to $n-1$) where $n = {}^sP_k$. It is an instance of a scheme for ranking k-permutations.

7.2. *Deep code as an extension/generalization of Lehmer code*

Deep code is similar to Lehmer code. Lehmer code is an example of an inversion table and is used to encode each possible permutation of a sequence of n numbers which encodes unique permutations. Lehmer code is a special case of Deep code. For $s = k$, and $n < s!$ Deep code is the same as Lehmer code. Deep code encoding scheme(also known as ranking) is explained in section (7.3) and Deep code decoding scheme(also known as un-ranking) is explained in section(7.5). Difference in the usage of Lehmer code and Deep code is depicted in the following Figure (2) -

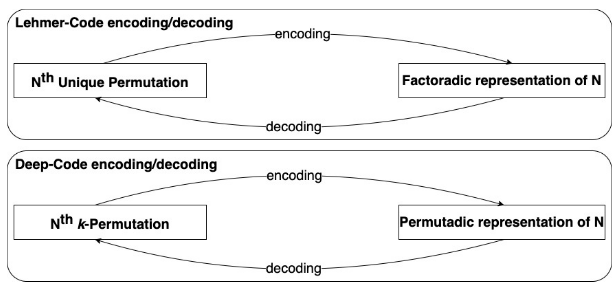

Figure 2: Lehmer Code Vs Deep Code encoding/decoding

7.3. *Encoding n^{th} k-permutation as permutadic*

Using the steps mentioned below, the n^{th} lexicographical k-permutation of s elements can be encoded as corresponding permutadic string.

- Let $[A_{k-1}, A_{k-2}, A_{k-3} \dots A_1, A_0]$ represents the indices of n^{th} lexicographical k-permutation from the indices $[0, 1, 2, \dots s-1]$ of s ordered elements.

- Create an ordered list L of all indices from 0 to $s-1$, which are not in $[A_{k-1}, A_{k-2}, A_{k-3} \dots A_1, A_0]$

- For i= 0 to $k-1$, calculate P[i] as follows -

  1. Find the index j in L where A_i can be inserted in ascending order
  2. Add A_i in L at j
  3. Calculate $P[i] = j$

- Return $[P_{k-1}, P_{k-2}, P_{k-3} \dots P_1, P_0][s]$ as encoded permutadic string

7.4. Deep code encoding example: encoding n^{th} k-permutation to permutadic

Let $A = [6, 0, 1, 4, 5]$ denotes the indices of n^{th} 5-permutation of 8 elements $S = [0, 1, 2, 3, 4, 5, 6, 7]$. To encode this to the corresponding permutadic string, create an ordered list $L = S - A = [2, 3, 7]$. Now find the index $j(2)$ in L where $A_0(5)$ can be inserted in ascending order. This gives $j = 2$. Inserting 5 in L gives $L = [2, 3, 5, 7]$. Calculate $P_0 = j(2)$. Now to find the values of P_1 to P_4, repeat the process. Value of L and P after each iteration is shown in the Table (5) below.

Table 5: Encoding $[6, 0, 1, 4, 5]$ as permutadic of size 8

$A[6, 0, 1, 4, 5]$	L	j	P
$A_0 = 5$	$[2, 3, 7]$	2	$[2]$
$A_1 = 4$	$[2, 3, 5, 7]$	2	$[2, 2]$
$A_2 = 1$	$[2, 3, 4, 5, 7]$	0	$[0, 2, 2]$
$A_3 = 0$	$[1, 2, 3, 4, 5, 7]$	0	$[0, 0, 2, 2]$
$A_4 = 6$	$[0, 1, 2, 3, 4, 5, 7]$	6	$[6, 0, 0, 2, 2]$
Output $= [6, 0, 0, 2, 2][8]$			

7.5. Decoding permutadic to n^{th} k-permutation:

Permutadic string $[C_{k-1}, C_{k-2}, \dots C_0][s]$ of size s and degree k can be decoded to n^{th} k-permutation if ${}^sP_k < n$. The decoding process is as follows -

- Create a list $S = [0, 1, 2, \dots s - 1]$
- For $i = k - 1$ to 1 do
  1. Set $P[i] = S[C[i]]$
  2. Remove $S[C[i]]$ from S
- Return P as the n^{th} permutation

7.6. Deep code decoding example: Decoding permutadic to n^{th} k-permutation

Suppose we need to decode the permutadic string $[6, 0, 0, 2, 2][8]$ to n^{th} 5-permutation. For this create a list $S = [0, 1, 2, 3, 4, 5, 6, 7]$. Now calculate $P[4] = S[[P[4]] = S[6] = 6$. Remove this value from S. So, $S = [0, 1, 2, 3, 4, 5, 7]$. Then calculate $P[3] = S[[P[3]] = S[0] = 0$ and again remove it from S. Repeat the process till $P[0]$. Table (6) depicts the steps for conversion of $[6, 0, 0, 2, 2][8]$ to 5-permutation of 8 elements.

Table 6: Decoding permutadic $[6,0,0,2,2][8]$ to n^{th} 5-permutation of 8 elements

P	S	$S[P[i]]$	Result
$P[4]=6$	$[0,1,2,3,4,5,6,7]$	$S[6]=6$	$[6]$
$P[3]=0$	$[0,1,2,3,4,5,7]$	$S[0]=0$	$[6,0]$
$P[2]=0$	$[1,2,3,4,5,7]$	$S[0]=1$	$[6,0,1]$
$P[1]=2$	$[2,3,4,5,7]$	$S[2]=4$	$[6,0,1,4]$
$P[0]=2$	$[2,3,5,7]$	$S[0]=5$	$[6,0,1,4,5]$
Output = [6, 0, 1, 4, 5]			

8. Permutational number system can be used to find n^{th} k-permutation

Proof: Let say we want to generate n^{th} k-permutations of s distinct elements where $1 \leq k \leq s$ and $0 \leq n < {}^{s}P_k$

We know that number of k-permutations from the list of s distinct elements is - ${}^{s}P_k$

The number of elements that can be selected at the first position(0^{th} index) is s,

$\therefore$ Each s element can be placed at first position for $\frac{{}^{s}P_k}{s} = {}^{s-1}P_{k-1}$ times.

$$\Rightarrow I_0 = \frac{n}{{}^{s-1}P_{k-1}} \tag{9}$$

Where I_0 is the index of first element out of s ordered elements for n^{th} permutation.

For the second position number of elements that can be selected is $s-1$ (as permutations cannot contain duplicates.)

$\therefore$ Each $s-1$ remaining elements can be placed at second position for $\frac{{}^{s-1}P_{k-1}}{s-1} = {}^{s-2}P_{k-2}$ times.

And the remaining value of n to be considered for the 2^{nd} position say n_1 is:

$$n_1 = n \pmod{{}^{s-1}P_{k-1}} \tag{10}$$

which gives,

$$I_1 = \frac{n_1}{^{s-2}P_{k-2}} \tag{11}$$

where I_1 is the index of the first element out of the remaining $s-1$ ordered elements. Using mathematical induction we can write Equation (10) and Equation (11) in general form as -

$$n_i = n_{i-1} \pmod{^{s-i}P_{k-i}} \tag{12}$$

and

$$I_i = \frac{n_i}{^{s-i-1}P_{k-i-1}} \tag{13}$$

where I_i is the index of the element from the remaining $s-i$ elements after selecting i elements.

from the definition of the permutational number system and Deep code, we can see that I_i is the i^{th} value from right in the permutadic representation of n so we can use permutadic representation to get the values from the said indices as per Deep code.

9. Putting everything together: A complete example

9.1. Finding lexicographic 1000^{th} 4-permutation of 8 elements:

- Bound check: As $1000 < {}^8P_4 = 1680$, hence it is possible to find the 1000^{th} 4-permutation of 8 elements.
- Find the permutadic representation of 1000 for size 8 and degree 4 as $[4, 5, 2, 0][8]$
- Decode $[4, 5, 2, 0][8]$ as per Deep code to get $[4, 6, 2, 0]$
- $[4, 6, 2, 0]$ is the lexicographical 1000^{th} 4-permutation of 8 elements, taking $[0, 1, 2, 3]$ as the 0^{th} permutation.

9.2. Converting 4-permutation $[4, 6, 2, 0]$ of 8 elements to decimal

- Using Deep code encoding, encode $[4, 6, 2, 0]$ to a permutadic of size 8 and degree 4 to get $[4, 5, 2, 0][8]$
- Convert permutadic $[4, 5, 2, 0][8]$ to decimal number system to get 1000
- $[4, 6, 2, 0]$ is the 1000^{th} lexicographical 4-permutation of 8-elements considering $[0, 1, 2, 3]$ as the 0^{th} 4-permutation.

10. Conclusion

- Any natural number can be represented in a permutational number system of size s and degree k, for some positive integers s and k where $1 \leq k \leq s$.
- A permutadic representation for a given (s, k, n) can be decoded to lexicographical n^{th} k-permutation of s elements if $n < {}^{s}P_{k}$ using Deep code decoding scheme.
- A given lexicographical n^{th} k-permutation of s elements can be encoded as permutadic representation of (s, k, n) using Deep code encoding scheme.
- Factorial number system is a special case of permutational number system and both representations are the same when $s = k$, and $n < s!$
- Lehmer code is a special case of Deep code and both encoding schemes are the same when $s = k$, and $n < s!$

11. Acknowledgment

I am thankful to my curious son Aditya Patel, who at the young age of 11 years, was curious to know how numbers are defined as permutations of symbols. My never ending discussions with him on the topic lead to the foundation of this paper.
I would also like to express my sincere gratitude to Pankaj Singh (linkedin.com /in/pankajsingh9), Rajeev Kumar (linkedin.com/in/rajeevkumar-a) and Osman Shoukry(linkedin.com/in/osmanshoukry/) for their valuable suggestions.

References

[1] D. E. Knuth, The art of computer programming, volume 4A: combinatorial algorithms, part 1, Pearson Education India, 2011.

[2] J. McCaffrey, Generating the mth lexicographical element of a mathematical combination, MSDN Library (2004).