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

Deepesh Patel[a], Aditya Patel[b]

[a]*Sr. Manager R&D, Sincro Digital Marketing, Lunkad Sky One Corporate Park, 3rd Floor, New Airport Road, Viman Nagar, Pune, 411014, MH, India*
[b]*Student at Vidya Pratishthan's Magarpatta City Public School, 51, Magarpatta, Hadapsar, Pune, 411028, 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 the 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 total number of permutations of 100 elements selected 50 at a time is $^{100}P_{50} = 3.068518756 \times 10^{93}$, 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

Contents

Email addresses: `patel.deepesh@gmail.com` (Deepesh Patel), `staradityapatel@gmail.com` (Aditya Patel)

Preprint submitted to Combinatorics, Probability and Computing *October 8, 2023*

Electronic copy available at: https://ssrn.com/abstract=4174035

Electronic copy available at: https://ssrn.com/abstract=4174035

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 a 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 a 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)

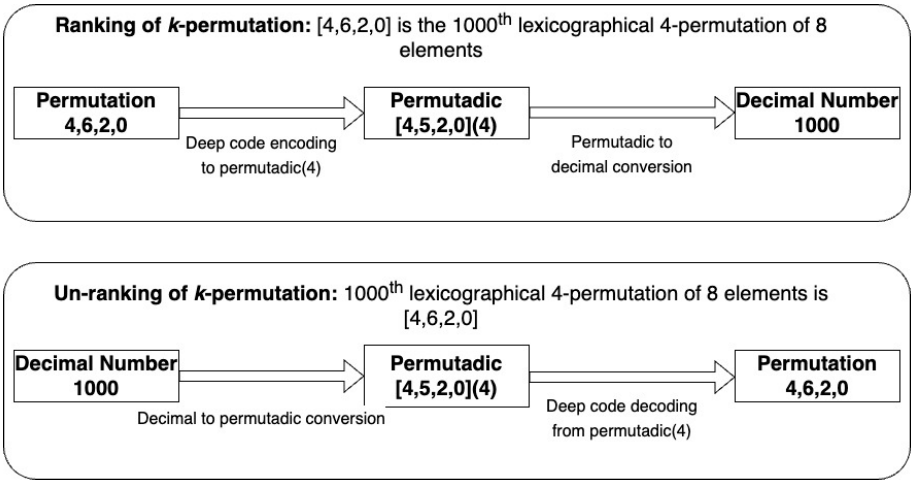

Figure 1: conversion of decimal number to/from k-permutation

2. Permutational Number System

2.1. Definition

Permutational number system of degree $d = s - k$, 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 of s items and is represented as a sequence $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0](d)$ where $C_i \in \mathbb{N}$.

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

$$[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0](d)$$

is expressed by the following equation -

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

Electronic copy available at: https://ssrn.com/abstract=4174035

where ${}^{s}P_k$ is the minimum number such that,

$$0 \leq n < {}^{s}P_k, \quad d = s - k$$

${}^{s-i}P_{k-i}$ is the place value for the i^{th} digit/number from right 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(3) as $[6, 0, 0, 2, 2](3)$ because -
$5050 = 6 \cdot {}^{7}P_4 + 0 \cdot {}^{6}P_3 + 0 \cdot {}^{5}P_2 + 2 \cdot {}^{4}P_1 + 2 \cdot {}^{3}P_0$

2.3. First 40 permutadic(2) numbers

Table (1) lists the decimal numbers and their permutadic(2) equivalent from 0 to 39. Place values(multiplier) of the digits are ${}^{4}P_2$, ${}^{3}P_1$ and ${}^{2}P_0$ from most significant value to least significant value, respectively. Leading zeros can be omitted.

Table 1: Permutadic(2) representation of integers from 0 to 39

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

2.4. Comparison of factoradic, combinadic and permutadic

Table (2) depicts the comparison between factoradic, combinadic, and permutadic. Permutadic can also be defined as having a fixed length like combinadic, but it is not in the scope of this paper.

Electronic copy available at: https://ssrn.com/abstract=4174035

Table 2: Comparison between number systems

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

3. Uniqueness of the permutadic representation (Proof of bijection)

Any radix (either standard or mixed) base representation of the natural number 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 t index is always the next weight minus one. It might not be clear at first sight in case of permutadic representation but can easily be proved with mathematical induction as explained in the following Theorem (1) -

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

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

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).

Electronic copy available at: https://ssrn.com/abstract=4174035

Also, from Equation (5), the highest value allowed for C_{k-i} is $(s - i)$ $\forall i | 1 \leq i \leq k$ because the highest possible value of the remainder is always one 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 correspondence between natural numbers and permutadic representation is bijective and we can say that there is a unique representation of all $n \in \mathbb{N}$ in a 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(d) of a natural number n, find the smallest s and k such that $s - k = d$ and ${}^{s}P_k \geq n$. (Larger values of s and n are also fine but will result in leading zeros). 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 ${}^{s-i}P_{k-i}$, $\forall\ ei \in \{x : x \in \mathbb{N}, 1 \leq x \leq k\}$

4.1. Algorithm - Convert decimal to permutadic(d):

I/P :- $s, k, n \mid d = s - k,\ {}^{s}P_k \geq n$

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 :- $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0](s - k)$

4.2. Optimized algorithm - Convert decimal to permutadic(d):

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

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

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

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

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

I/P :- n, d
$j = d + 1$
$i = 0$
do

- $C_i = n \pmod{j}$
- $n = n/j$
- $j = j + 1$
- $i = i + 1$

while $n > 0$
O/P :- $[C_{k-1}, C_{k-2}, C_{k-3} \ldots C_1, C_0](d)$

Electronic copy available at: https://ssrn.com/abstract=4174035

4.3. Example:

Table (3) depicts the conversion of 5050 to a permutational number system of degree 3

Table 3: Conversion of decimal integer 5050 to Permutadic(3)

n	Divisor	Quotient	Remainder
5050	$j = d + 1 = 4$	$5050/4 = 1262$	2
1262	5	$1262/5 = 252$	2
252	6	$252/6 = 42$	0
42	7	$42/7 = 6$	0
6	8	0	6
Output = $[6, 0, 0, 2, 2](3)$			

5. Conversion from permutational number system to decimal

To convert the permutational number $[C_{k-1}, C_{k-2}, C_{k-3} \dots C_1, C_0](d)$ 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.

5.1. Algorithm: Convert permutadic to decimal

I/P :- $[C_{k-1}, C_{k-2}, C_{k-3} \dots C_1, C_0](d)$
$sum = C_0$
$counter = d + 1$
$multiplier = counter$
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 $[6, 0, 0, 2, 2](3)$ to decimal equivalent

Table 4: Conversion of permutadic [6, 0, 0, 2, 2](3) to decimal

C_i	*counter*	*multiplier*	*sum*
2	NA	1	$2 \cdot 1 = 2$
2	$d + 1 = 4$	$1 \cdot 4 = 4.$	$2 + (2 \cdot 4) = 10$
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			

Electronic copy available at: https://ssrn.com/abstract=4174035

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

6.1. Permutadic of degree 0

For the permutational number system of degree 0, the representation will be the same as the factorial number system after omitting the degree from the representation. This is explained in the Theorem (2) below.

Theorem 2. *The permutational number system of degree* 0 *contains the same sequence of numbers as in factorial number system* $\forall n \in \mathbb{N}$

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

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

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

Which denotes the n^{th} natural number in the factorial number system.

6.2. Example:

Table (5) represents the factoradic and permutadic(0) representation of numbers between 5050 and 5058

Table 5: Permutadic(0) and factoradic representation of integers from 5050 to 5058

Decimal	Permutadic(0)	Factoradic
5050	[1, 0, 0, 0, 1, 2, 0, 0](0)	[1, 0, 0, 0, 1, 2, 0, 0]
5051	[1, 0, 0, 0, 1, 2, 1, 0](0)	[1, 0, 0, 0, 1, 2, 1, 0]
5052	[1, 0, 0, 0, 2, 0, 0, 0](0)	[1, 0, 0, 0, 2, 0, 0, 0]
5053	[1, 0, 0, 0, 2, 0, 1, 0](0)	[1, 0, 0, 0, 2, 0, 1, 0]
5054	[1, 0, 0, 0, 2, 1, 0, 0](0)	[1, 0, 0, 0, 2, 1, 0, 0]
5055	[1, 0, 0, 0, 2, 1, 1, 0](0)	[1, 0, 0, 0, 2, 1, 1, 0]
5056	[1, 0, 0, 0, 2, 2, 0, 0](0)	[1, 0, 0, 0, 2, 2, 0, 0]
5057	[1, 0, 0, 0, 2, 2, 1, 0](0)	[1, 0, 0, 0, 2, 2, 1, 0]
5058	[1, 0, 0, 0, 3, 0, 0, 0](0)	[1, 0, 0, 0, 3, 0, 0, 0]

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 $n = {}^{s}P_k$. It is an instance of a scheme for ranking k-permutations.

7.2. Deep code as a generalization of Lehmer code

Deep code is a generalization of 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 $s = k$, that is for degree 0, the Deep code is the same as the Lehmer code. The Deep code encoding scheme (also known as ranking) is explained in section (7.3) and the Deep code decoding scheme (also known as un-ranking) is explained in section(7.5). The difference in the usage of Lehmer code and Deep code is depicted in the following Figure (2) -

Electronic copy available at: https://ssrn.com/abstract=4174035

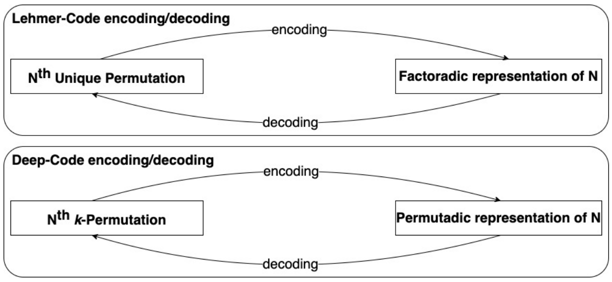

Figure 2: Lehmer Code Vs Deep Code encoding/decoding

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 a 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, i.e, $j : L[j] < A[i] < L[j+1]$
  2. insert A_i in L at j
  3. set $P[i] = j$
- return $[P_{k-1}, P_{k-2}, P_{k-3} \dots P_1, P_0](s-k)$ 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 (6) below.

Table 6: Encoding [6, 0, 1, 4, 5] as permutadic of degree 3

$A[6, 0, 1, 4, 5]$	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](3)			

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

Permutadic string $[C_{k-1}, C_{k-2}, \dots C_0](d)$ of degree d can be decoded to n^{th} k-permutation if ${}^sP_k < n$ and $d = s - k$. To avoid complexity, the decoding process mentioned below assumes that the permutadic string is of size k and contains leading zeroes if required to make up for the desired size. The algorithm for decoding is as follows -

- Create a list $S = [0, 1, 2, \dots s-1]$

Electronic copy available at: https://ssrn.com/abstract=4174035

- j=0
- For $i = k-1$ to 0 do
  1. Set $P[j] = S[C[i]]$
  2. Remove $S[C[i]]$ from S
  3. j=j+1
- 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 $[6, 0, 0, 2, 2](3)$ to n^{th} 5-permutation of 8 items. 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 (7) depicts the steps for conversion of $[6, 0, 0, 2, 2](3)$ to 5-permutation of 8 elements.

Table 7: Decoding permutadic $[6, 0, 0, 2, 2](3)$ 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: Suppose 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 the 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 the 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,

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

Electronic copy available at: https://ssrn.com/abstract=4174035

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(4) representation of 1000 as $[4, 5, 2, 0](4)$
- Decode $[4, 5, 2, 0](4)$ as per Deep code to get $[4, 6, 2, 0]$
- 1000^{th} lexicographical 4-permutation of 8 elements is $[4, 6, 2, 0]$, considering $[0, 1, 2, 3]$ as the 0^{th} permutation.

9.2. Finding the rank of 4-permutation $[4, 6, 2, 0]$ of 8 elements

- Using Deep code encoding, encode $[4, 6, 2, 0]$ to a permutadic of degree 4 to get $[4, 5, 2, 0](4)$
- Convert permutadic $[4, 5, 2, 0](4)$ 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 degree d, where $d \in \mathbb{N}$
- A permutadic representation for a given (s, k, n) can be decoded to lexicographical n^{th} k-permutation of s elements if $n < {}^sP_k$ 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 for $s = k$.
- Lehmer code is a special case of Deep code and both encoding schemes are the same for $s = k$.

11. Acknowledgment

I am thankful to my curious son Aditya Patel, a co-author of this paper, 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 led to the foundation of this paper. I am also thankful to him for suggesting a crucial optimization in the algorithm used in 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.

Electronic copy available at: https://ssrn.com/abstract=4174035

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).

Electronic copy available at: https://ssrn.com/abstract=4174035