Using Generating Functions on the Maximum Number of Inversions in *k-sorted* Permutations with Weakly Restricted Displacements

Ting Kuo
Takming University of Science and Technology
tkuo@takming.edu.tw

Abstract

A permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$ of size n is called *k-sorted* if and only if $|i - \pi_i| \le k$, for all $1 \le i \le n$. We propose an algorithm for generating the set of all *k-sorted* permutations of size n in *lexicographic* order. An inversion occurs between a pair of (π_i, π_j) if $i < j$ but $\pi_i > \pi_j$. The maximum number of inversions in *k-sorted* permutations of size n, denoted by $I(n, k)$, is a measure of disordered in permutations. By using the *generating functions* approach, we obtain a formula, $I(n,k) = 2kn - k(k+1) - n(n-1)/2$, for permutations with *weakly* restricted displacements, i.e., $\lfloor n/2 \rfloor \le k \le n-1$.

Keywords: k-sorted permutations; generating functions; inversions; lexicographic order; ordinal representation

1. Introduction

The goal of this paper is twofold. First, we will propose an algorithm for generating the set of all *k-sorted* permutations of size n in *lexicographic* order. Second, we will propose a formula of maximum number of inversions in *k-sorted* permutations of size n with weakly restricted displacements. As a beginning, we use notation S_n to denote the set of all permutations of size n. That is, a permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$ belongs to S_n if and only if

$$\pi_i \in \{1,\ldots,n\}, \text{ for all } i = 1,\ldots,n,$$

and

$$\pi_i \ne \pi_j, \text{ for all } i \ne j.$$

For a permutation $\boldsymbol{\pi}$, if it satisfies $\pi_i = i$ then we say that integer i is located at its *right* place. In this paper, we focus on a special kind of permutations, called *k-sorted* permutations, that every integer i is at most k displacements from its right place. We use notation K_n to denote the set of all *k-sorted* permutations of size n that satisfies the following definition [3].

Definition 1. A permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$ is called *k-sorted* if and only if

$$|i - \pi_i| \le k, \text{ for all } 1 \le i \le n.$$

It is obvious that $0 \le k \le n-1$ and that $K_n \subseteq S_n$. Notice that the numbers of n and k are positive integers throughout this paper.

Few studies have been devoted to this topic. Bongiovanni *et al*. [14] introduced a problem of *quasi sorting* as the one of transferring a given permutation to a *k-sorted* permutation. In that paper, he only proposed the bases for developing algorithms for generating *k-sorted* permutations through element comparisons. Besides, although many algorithms have been devoted for generating S_n and various permutation problems [6, 13], we know of no published algorithms for generating K_n in *lexicographic* order. Therefore, we will propose an algorithm for generating the set of all *k-sorted* permutations of size n in *lexicographic* order.

Just as *permutations* play an important role in the study of combinatorics, *inversions* of a permutation is crucial in the study of permutations [3, 5, 11]. Now, let's introduce some definitions that are concerned about inversion.

Definition 2. In a permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$, an inversion occurs between a pair of (π_i, π_j) if $i < j$ but $\pi_i > \pi_j$.

Definition 3. Let $I(\pi_j)$ denotes the number of inversions of π_j, then $I(\pi_j)$ is the number of k's such that $j < k$ but $\pi_j > \pi_k$.

Definition 4. Let $I(\boldsymbol{\pi})$ denotes the number of inversions in a permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$, then

$$I(\pi) = \sum_1^n I(\pi_j).$$

It is well known that the value of $I(\boldsymbol{\pi})$ is a measure of disordered in a permutation $\boldsymbol{\pi} = (\pi_1\pi_2\ldots\pi_n)$. In this study, we focus on the maximum number of inversions in K_n.

Definition 5. Let $I(n, k)$ denotes the maximum number of inversions in *k-sorted* permutations of size n, that is

$$I(n,k) = \max\{I(\pi), \text{for all } \pi \in K_n\}.$$

Berman [1] gave a value of $2kn$ as an upper bound of $I(n, k)$. Dutton [3] improved the upper bound of $I(n, k)$ down to a value of $0.6kn$ by proposing a sophisticated formula as follows

$$I(n,k) = 2kn - \min\{f(t_1), f(t_2)\},$$

with

$$t_1 = \left\lfloor \frac{n}{m} \right\rfloor, t_2 = \left\lceil \frac{n}{m} \right\rceil, m = \left\lceil \frac{-1+\sqrt{8k^2+8k+1}}{2} \right\rceil,$$

and the function $f(t)$ is defined as follows

$$f(t) = t\left\lceil k(k+1) - \left\lfloor \frac{n}{t} \right\rfloor \frac{\left\lfloor \frac{n}{t} \right\rfloor + 1}{2} \right\rceil + n\left\lfloor \frac{n}{t} \right\rfloor. \quad (1)$$

Here, $\lfloor x \rfloor$ (read "the floor of x") stands for the greatest integer less than or equal to x, and $\lceil x \rceil$ (read "the ceiling of x") the least integer greater than or equal to x. In this paper, for permutations with weakly restricted displacements, i.e., $\lfloor n/2 \rfloor \le k \le n-1$, we will propose a concise formula as follows

$$I(n,k) = 2kn - k(k+1) - n(n-1)/2. \quad (2)$$

We borrow the name "permutations with *weakly* restricted displacements" from Lehmer [9]. In that paper he proposed generating functions for *k-sorted* permutations, with $k = 1,2,3$, that he called permutations with *strongly* restricted displacements.

The rest of this paper is organized as follows. In Section 2, we talk about representation schemes for permutations. In Section 3, we describe an algorithm that is used to generate, in *lexicographic* order, K_n and to compute $I(n, k)$. In Section 4, we present some examples of $I(n, k)$'s that are used in the following section. In Section 5, we derive the formula of $I(n, k)$ by using the *generating function* approach. Conclusions are summarized in Section 6.

2. Ordinal Representation Scheme

In combinatorics and mathematics, several representation schemes have been used for permutation, such as *two-line* form [4, p. 160], *cycle* notation [4, p. 160], *permutation matrix* [2, p.728], *inversion vector* [12, p.164], *inversion table* [5, pp. 11]. Previously, from a different operational point of view, we proposed a new representation scheme of permutation, called *ordinal representation* [7]. Now, let's take a briefly look at it.

Definition 6: For a permutation $\boldsymbol{\pi}$ in the form of *ordinal representation*, that is $[D_n D_{n-1} \cdots D_1]$, $\boldsymbol{\pi}$ belongs to S_n if and only if

$$1 \le D_j \le j, \quad \text{for all} \quad j = 1,2,\ldots,n.$$

We call $[D_n D_{n-1} \cdots D_1]$ the *ordinal digits* of a permutation $\boldsymbol{\pi}$.

The meaning of *ordinal digits* is easy to understand if we imagine a permutation as the result of a successive withdrawing of items individually, one after the other without replacement, from an ordered item set $\{1,\ldots,n\}$. At the beginning, there are n choices we can choose to be the first component of $\boldsymbol{\pi}$. That is why we let $1 \le D_n \le n$. Once we have chosen an item as the first component of $\boldsymbol{\pi}$, there are $n-1$ choices left in the ordered item set. So, we let $1 \le D_{n-1} \le n-1$. In the end, only one choice left, so we let $1 \le D_1 \le 1$. In other words, the component π_{n-j+1} of $\boldsymbol{\pi}$ is determined by D_j. Furthermore, the value of D_j is one plus the number of items that are less than π_{n-j+1} and to the right of it.

Since each permutation $\boldsymbol{\pi}$ in S_n corresponds uniquely to an integer q in the range of $[0, n!-1]$, we have the following theorem.

Theorem 1. In S_n, there is a *one-to-one* correspondence between $[D_n D_{n-1} \cdots D_1]$ and $(\pi_1 \pi_2 \ldots \pi_n)$.

Proof. Clearly, it is easy to convert an integer q to its *factorial representation* [8, p. 7]. First, we divide the integer q by $(n-1)!$ and set the quotient to C_{n-1}, then the remainder is divided by $(n-2)!$ and the quotient is set to C_{n-2}, and so on. That is, any integer q between 0 and $n!-1$ can be represented as

$$q = C_{n-1} \times (n-1)! + C_{n-2} \times (n-2)! + \cdots + C_1 \times 1! + C_0 \times 0!. \qquad (3)$$

Here, the following constraints

$$1 \le C_i \le i, \quad \text{for all} \quad i = 0,1,\ldots,n-1,$$

are imposed to ensure uniqueness. These C_i's are called the *factorial digits* of integer q [8, p. 7]. By Definition 6, we know that

$$1 \le D_j \le j, \quad \text{for all} \quad j = 1,2,\ldots,n.$$

Hence, we have a *one-to-one* correspondence between D_j and C_i as follows:

$$D_j = C_i + 1, \quad \text{where} \quad j = i + 1,$$

for all $j = 1,2,\ldots,n.$ ■

Thus, if we order all permutations of S_n in *lexicographic* order then we can, for example when n = 7, use the *ordinal digits* [1 1 1 1 1 1 1] to represent the first (i. e., 0[th]) permutation $\boldsymbol{\pi}$ = (1 2 3 4 5 6 7) and [7 6 5 4 3 2 1] to the last (i. e., 5039[th]) permutation $\boldsymbol{\pi}$ = (7 6 5 4 3 2 1), respectively. It is easy to see that $D_n = \pi_1$ for all permutations of S_n.

3. Algorithm

Although many algorithms have been devoted for generating S_n and various permutation problems [6, 13], we know of no published algorithms for generating K_n in *lexicographic* order. In a previous study [7], by using *ordinal representation*, we proposed a method for generating permutations in *lexicographic* order. In this study, we extend this method to generate K_n in *lexicographic* order and compute $I(n, k)$. The extension is based on an interesting property that the number of inversions occur in a permutation is equal to the summation of its *ordinal digits* minus n. This property is demonstrated in the following theorem.

Lemma 1. For a permutation $\boldsymbol{\pi}$ in the form of *ordinal representation*, $\boldsymbol{\pi} = [D_n D_{n-1} \cdots D_1]$, we have $D_{n-j+1} = I(\pi_j) + 1$, for all $j = 1,2,\ldots,n$.

Proof. By Definitions 3, we know that $I(\pi_j)$ is the number of k's such that $j < k$ but $\pi_j > \pi_k$. From the meaning of *ordinal digits* mentioned above, we know that the value of D_j is one plus the number of items that are less than π_{n-j+1} and to the right of it. In other words, we have $D_j = I(\pi_{n-j+1}) + 1$.

That is, $D_{n-j+1} = I(\pi_j) + 1$. ■

Theorem 2. For a permutation $\boldsymbol{\pi}$ in the form of *ordinal representation*, $\boldsymbol{\pi} = [D_n D_{n-1} \cdots D_1]$, we have $I(\pi) = \sum_1^n D_j - n$.

Proof. By Definition 4 and Lemma 1, we have $I(\pi) = \sum_1^n I(\pi_j) = \sum_1^n D_j - n$. ■

Therefore, by using this *ordinal representation* scheme, we can systematically generate the whole K_n in lexicographic order and output $I(n, k)$. This task can be described as the following algorithm.

Algorithm 1. Generate K_n in lexicographic order and compute $I(n, k)$.

Input: n and k.

Output: K_n and $I(n, k)$.

```
Begin
 I(n, k) = 0
 For D_n = 1 To n
  For D_(n-1) = 1 To n – 1
        …
     For D_1 = 1 to 1
       Let item set A = {1,…, n}
       For i = n To 1
         Retrieve the D_i th item of A
         If the D_i th item satisfies Def. 1 then
           Let π_(n-i+1) = the D_i th item of A
           Delete the D_i th item of A
           Goto Next i
         Else
           Select case i
             Case n
               Goto Next D_n
             Case n – 1
               Goto Next D_(n-1)
                 …
             Case 1
               Goto Next D_1
           End Select
         Endif
       Next i
       Output π = (π_1π_2…π_n) and Compute I(π)
       If I(π) > I(n, k) Then Let I(n, k) = I(π)
     Next D_1
     …
  Next D_(n-1)
 Next D_n
 Output I(n, k)
End
```

4. Examples of *I*(n, k)'s

In this section, we present some examples of $I(n, k)$'s in Table 1. By carefully observing the numbers in Table 1, we find the following recurrences which are corresponding to the diagonals of Table 1. For convenience, we use "the i^{th} diagonal of Table 1" to stand for those numbers that are presented in the i^{th} diagonal of

Table 1.

In case of the first diagonal of Table 1, i.e., $k = n - 1$, for $n \geq 2$ we have the following recurrence

$$I(n, n-1) = I(n-1, n-2) + n - 1, \quad (4)$$

with $I(1,1) = 0$.

In case of the second diagonal of Table 1, i.e., $k = n - 2$, for $n \geq 4$ we have the following recurrence

$$I(n, n-2) = I(n-1, n-3) + n - 1, \quad (5)$$

with $I(3,1) = 1$.

In case of the third diagonal of Table 1, i.e., $k = n - 3$, for $n \geq 6$ we have the following recurrence

$$I(n, n-3) = I(n-1, n-4) + n - 1, \quad (6)$$

with $I(5,2) = 4$.

Now, it is not difficult to generalize these recurrences as follows.

First, for $n = 2r + 1$, with integer $r \geq 0$, we have

$$I(2r+1, r) = r^2. \quad (7)$$

Second, for $n \geq 2r + 2$, with integer $r \geq 0$, we have

$$I(n, n-r-1) = I(n-1, n-r-2) + n - 1. \quad (8)$$

Although we have found these recurrences, we still are not satisfied. Our goal is to find an explicit closed formula that can give us an answer quickly. It is discussed in next section.

5. A Formula of *I*(n, k)

In this section, for permutations with weakly restricted displacements, that is $\text{int}(n/2) \leq k \leq n-1$, we will derive a concise formula of $I(n, k)$ by using the *generating function* approach. Usually, a *generating function* is a power series. The two most common types of generating functions are *ordinal generating functions* and *exponential generating functions* [10]. The result is the following theorem.

Theorem 3. For $\text{int}(n/2) \leq k \leq n-1$, with $n \geq 2$, the maximum number of inversions in *k-sorted* permutations of size n is

$$I(n,k) = 2nk - k(k+1) - \frac{n(n-1)}{2}. \quad (9)$$

Proof. It is easy to discover the explicit closed formula for the first diagonal of Table 1, i.e., $k = n-1$. By Definition 3, we should put the larger number as left as possible and put the smaller number as right as possible. Since all permutations in S_n are $(n-1)$ *-sorted*, the maximum number of inversions occurs in the last permutation $\boldsymbol{\pi} = (n \ \ n-1 \ldots 2 \ 1)$. Thus,

$$I(n, n-1) = \frac{n(n-1)}{2}. \quad (10)$$

So, we start by discussing the second diagonal of Table 1. First, let $A(x)$ denotes the *ordinal generating function* of the second diagonal of Table 1, i.e., $k = n - 2$, the sequences $(a_3, a_4, a_5, \ldots, a_n, \ldots)$, as follows

$$A(x) = a_3 + a_4 x + a_5 x^2 + \cdots + a_n x^n + \cdots.$$

Then, we rewrite the corresponding recurrence (5) as

$$a_n = a_{n-1} + n - 1, \text{for } n \geq 4, \text{with } a_3 = 1. \quad (11)$$

To find $A(x)$, we multiply both sides of the recurrence (11) by x^n and sum over $n \geq 4$, then we have

$$\sum_{n=4}^{\infty} a_n x^n = \sum_{n=4}^{\infty} a_{n-1} x^n + \sum_{n=4}^{\infty} (n-1) x^n.$$

That is,

$$(A(x) - a_3)x^3 = A(x)x^4 + \frac{x^2}{(1-x)^2} - 2x^3.$$

Since $a_3 = 1$, we obtain

$$A(x)(x^3 - x^4) = \frac{x^2}{(1-x)^2} - x^2.$$

Thus, we have

$$A(x) = \frac{1 + x - x^2}{(1 - x)^3}. \quad (12)$$

In order to find an explicit formula for the sequences (a_3, a_4, a_5, …, a_n,…), we have to expand $A(x)$ in a series of partial fraction. Fortunately, it is easy to expand $A(x)$ as follows.

$$\begin{aligned} A(x) &= \frac{2}{(1-x)^3} - \frac{1}{(1-x)^2} - \frac{x^2}{(1-x)^3} \\ &= 2\sum_{n=0}^{\infty}\binom{n+2}{2}x^n - \sum_{n=0}^{\infty}\binom{n+1}{1}x^n - \sum_{n=0}^{\infty}\binom{n}{2}x^n \\ &= \sum_{n=0}^{\infty}\left(\frac{n^2+5n+2}{2}\right)x^n. \quad (13) \end{aligned}$$

Here, the coefficient of x^0, i.e., a_3, equal to 1, is the value of $I(3, 1)$; the coefficient of x^1, i.e., a_4, equal to 4, is the value of $I(4, 2)$; the coefficient of x^2, i.e., a_5, equal to 8, is the value of $I(5, 3)$; and so on.

Similarly, let $B(x)$ denote the *ordinal generating function* of the third diagonal of Table 1, i.e., $k = n - 3$, the sequences (b_5, b_6, b_7, …, b_n,…), as follows

$$B(x) = b_5 + b_6 x + b_7 x^2 + \cdots + b_n x^{n-5} + \cdots.$$

The corresponding recurrence (6) can be rewritten as

$$b_n = b_{n-1} + n - 1, \text{ for } n \geq 6, \text{ with } b_5 = 4.$$

As before, we have

$$B(x) = \frac{4 - 3x}{(1-x)^3}. \quad (14)$$

After expand $B(x)$ in a form of partial fraction, we obtain an explicit formula for the sequences (b_5, b_6, b_7, …, b_n,…) by

$$B(x) = \sum_{n=0}^{\infty}\left(\frac{n^2+9n+8}{2}\right)x^n. \quad (15)$$

Here, the coefficient of x^0, i.e., b_5, equal to 4, is the value of $I(5, 2)$, the coefficient of x^1, i.e., b_6, equal to 9, is the value of $I(6, 3)$, the coefficient of x^2, i.e., b_7, equal to 15, is the value of $I(7, 4)$, and so on.

Hence, in general, let $F(x)$ denote the *ordinal generating function* of the a^{th} diagonal of Table 1, i.e., $k = n - a$, we have

$$F(x) = \sum_{n=0}^{\infty}\left(\frac{n^2 + (4a-3)n + 2(a-1)^2}{2}\right)x^n. \quad (16)$$

Here, the coefficient of x^i is the value of

$$I(2a - 1 + i, a - 1 + i).$$

For example, what is the maximum number of inversions in 6-*sorted* permutations of size nine? First, since $a = n - k = 3$, we know that $I(9,6)$ is in the third diagonal of Table 1. It is note that the first number in the third diagonal of Table 1 is the coefficient of x^0. So, by $i = n - 2a + 1 = 4$, or by $i = k - a + 1 = 4$, we know that $I(9,6)$ is the fifth number in the third diagonal of Table 1. Therefore,

$$\begin{aligned} I(n,k) &= I(9,6) \\ &= \frac{4^2 + (4\times 3 - 3)\times 4 + 2(3-1)^2}{2} = 30. \end{aligned}$$

Since $a = n - k$, we have $i = 2k - n + 1$. Thus, replace a by $n - k$, and n by $2k - n + 1$, we finally have

$$I(n,k) = 2nk - k(k+1) - \frac{n(n-1)}{2}. \quad \blacksquare$$

6. Conclusions

The algorithm we proposed is quite suit for generating the set of all *k-sorted* permutations of size n in lexicographic order. By using the *generating functions* approach, we obtain a formula, $I(n,k) = 2kn - k(k+1) - n(n-1)/2$, to determine the maximum number of inversions for permutations with weakly restricted displacements. The beauty of the *generating function* approach lies not in the result itself, but rather in its wide applicability.

References

1. K. A. Berman, J. L. Paul, **Fundamentals of Sequential and Parallel Algorithms**, PWS Publishing Co., Boston, MA, 1997.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, **Introduction to Algorithms**, Second Edition, The MIT Press, 2001.
3. Ronald D. Dutton, Inversions in k-sorted Permutations, **Discrete Applied Mathematics**, 87 (1998), pp. 49-56.
4. D. E. Knuth, **The Art of Computer Programming, Volume 1: Fundamental Algorithms**, Second Edition, Addison-Wesley, 1973.
5. D. E. Knuth, **The Art of Computer Programming, Volume 3: Sorting and Searching**, Second Edition, Addison-Wesley, 1998.
6. D. E. Knuth, **The Art of Computer Programming, Volume 4, Fascicle 2: Generating all Tuples and Permutations**, Addison-Wesley, 2005.
7. T. Kuo, A New Method for Generating Permutations in Lexicographic Order, submitted.
8. D. H. Lehmer, The Machine Tools of Combinatorics, in **Applied Combinatorial Mathematics**, E. F. Beckenbach, ed., John Wiley & Sons, Inc., N Y, 1964, pp. 5-31.
9. D. H. Lehmer, Permutations with strongly restricted displacements, in: **Combinatorial Theory and Its Applications** (Balatonfüred: Hungary, 1969), II, Colloq. Math. Soc. János Bolyai 4, edited by P. Erdòs, A. Rényi and Vera T. Sós, North-Holland, Amsterdam 1970, pp. 755–769.
10. C. L. Liu, **Introduction to Combinatorial Mathematics**, Mcgraw-Hill College, 1968.
11. Barbara H. Margolius, Permutations with Inversions, **Journal of Integer Sequences**, Vol. 4 (2001), Article 01.2.4.
12. E. M. Reingold, J. Nievergelt, and N. Deo, **Combinatorial Algorithms: Theory and Practice**, Prentice-Hall, Inc., 1977.
13. R. Sedgewick, Permutation Generation Methods, **ACM Computing Surveys**, Vol. 9, No. 2 (1977), pp. 137-163.
14. G. Bongiovanni, F. Luccio and L. Pagl, The problem of quasi sorting, **Calcolo**, Volume 16, Number 4 (1979), pp. 415-430.

Table 1. $I(n, k)$, **for** $2 \le n \le 15$, **and** $\lfloor n/2 \rfloor \le k \le n-1$.

n \ k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1													
3	1	3												
4		4	6											
5		4	8	10										
6			9	13	15									
7			9	15	19	21								
8				16	22	26	28							
9				16	24	30	34	36						
10					25	33	39	43	45					
11					25	35	43	49	53	55				
12						36	46	54	60	64	66			
13						36	48	58	66	72	76	78		
14							49	61	71	79	85	89	91	
15							49	63	75	85	93	99	103	105