The Perfect Algorithm for Generating Permutation

Lugen M. Zake[1], Haslinda Ibrahim[2], Zurni Omar[3]

[1]Department of Mathematics, College of Basic Education, University of Mosul, Iraq
Mosul, Iraq
[2,3]School of Quantitative Sciences, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah
Malaysia

Abstract: The most difficult task dealing with permutation is when the element is large. In this paper, a new algorithm for generating permutation based on fixing two elements. The basic idea for this method is to find original and sequent starter set from fixing one element from the first and one element from the last elements to begin with as an opening set to generate all different permutations, in circular operations is presented. Beginning the original starter sets are obtained by fixing two elements, and then fixing another element from last to find sequent starter sets, the circular operation and reverse are in use to produce different permutation from each sequent starter sets. The new algorithm has advantages over the other methods due to its simplicity and easy to use.

Keywords: Permutation, algorithm permutation, starter set permutation.

Introduction

A permutation of elements is an arrangement of those elements in some order, that is, some element is located in the first location, another in the second location, and so on, until the last element have been last located. Since the generation of all n! permutations of n elements are a fundamental problem in combinatorics and important in computing. Therefore most of the existing algorithms essence on listing all n! permutations. So various methods on listing all permutations have been published by Sedgeweck (1977) and can be classified into two categories: (i) exchange based techniques; (ii) non-exchange based techniques (Sedgeweck, 1977). So generating permutation under cycling restriction is simpler and more powerful compared to others above- mentioned techniques. Therefore Langdon (1967) and Iyer (1995) proposed a cycling technique for permutation generation based on 'pivot' permutation. Akl (1980) found a new algorithm for generating permutation (Akl, 1980). So Bauslaugh and Ruskey published algorithm for generating alternating permutations lexicographical (Bauslaugh & Ruskey, 1990). And Pruess and Ruskey published algorithm for generated permutations for linear extension of posets (Pruess & Ruskey, 1994). Kunth summarized nicely the computational and mathematical properties of permutations with a given number of inversions (Knuth, 1998). Dominique (1999) presented a new algorithm for generating of permutations for listing all permutations of the integers 1, 2,… , n. Also Effler and Ruskey (2003) published algorithm for generated permutations with a given number of inversions. Andrew King (2006) presented algorithm for generating of permutations in decomposable permutations. Khanit (2007) found a method to generate permutation and he used it to find determinant of any matrix. Meanwhile Ibrahim, et. (2010) and Ibrahim, & Zake (2013) introduced a new permutation technique based on distinct starter sets by employing circular and reversing operations. However their crucial task is to generate the distinct starter sets only. Although this technique is simple and easy to use, unfortunately, finding the starter sets is quite tedious when the number of elements increases. Thus this paper aims to provide a method to list n! permutations based on original starter set with fixing two elements. This paper attempts to overcome this drawback by introducing a new strategy for generating distinct original starter sets with fixing two elements one from the first and one from the last elements. The advantage of this proposed algorithm is to produce some sets to begin with and these sets finally generate all the n! permutations. To our present knowledge no study has been done using this kind of algorithm.

Preliminary Definition

Definition 1: A starter set is a set that is used as a basis to enumerate other permutations.

Definition 2: A original starter sets are a sets which result by fixing two elements and used as to create other permutations.

Definition 3: An equivalence original starter sets are sets that can produce by taking inverse the same elements permutation after fixing two elements first and last.

Definition 4: A Sequent starter sets are a sets which result from original starter sets by fixing two elements also but the same first element and used as to create other permutations.

Definition 5: A reversed (inverse) set is a set that is produced by reversing the order of permutation set.

Implementation of Perfect Permutation Algorithm

Before The general algorithm for generating permutation as follows, let S be the set of *n* elements i.e. (1, 2, 3, …, *n*-3, *n*-2, *n*-1, *n*), Set (1, 2, 3, 4, …, *n*-3, *n*-2, *n*-1, *n*) as initial permutation and without loss of generality.

Step1/ Find original starter sets permutation by fixing two elements, fixing the first and the last elements of the initial permutation to generated original starter sets, then move and exchange remaining elements without repeating, then delete the permutations that produce the inverse (reverse) switched elements. See in the following

$$(1\ \ 2\ \ 3… n\text{-}2\ \ n\text{-}1\ \ n) \cong (1\ \ n\text{-}1\ \ n\text{-}2 … 3\ \ 2\ \ n)$$
$$(1\ \ 2\ \ 3… n\text{-}1\ \ n\text{-}2\ \ n) \cong (1\ \ n\text{-}2\ \ n\text{-}1 … 3\ \ 2\ \ n)$$
$$(1\ \ 2\ \ 3… n\text{-}1\ \ n\text{-}3\ \ n) \cong (1\ \ n\text{-}3\ \ n\text{-}1 … 3\ \ 2\ \ n)$$
$$\vdots$$
$$(1\ \ n\text{-}1\ \ n\text{-}2… 3\ \ 2\ \ n) \cong (1\ \ 2\ \ 3 … n\text{-}2\ \ n\text{-}1\ \ n)$$

So we choose only one group of permutations and delete equivalents inverse group of permutations these group of permutation called original starter sets.

Step2/ Find sequent starter sets of permutation by fixing two elements of all permutation resulted in the step 1 and also fixing the same first element and another the last element and then make exchange to the remaining elements after fixing and then delete permutations equivalent (inverse permutation without machineries fixing).
This step is similar to the first step by fixing two elements but on original starter sets which is resulted in step 1 as follows:-

$$(1\ \ 2\ \ 3… n\text{-}2\ \ n\text{-}1\ \ n) \cong (1\ \ n\text{-}1\ \ n\text{-}2 … 3\ \ 2\ \ n)$$
$$(1\ \ 2\ \ 3… n\text{-}2\ \ n\ \ n\text{-}1) \cong (1\ \ n\ \ n\text{-}2 … 3\ \ 2\ \ n\text{-}1)$$
$$(1\ \ 2\ \ 3… n\text{-}3\ \ n\ \ n\text{-}1) \cong (1\ \ n\ \ n\text{-}3 … 3\ \ 2\ \ n\text{-}1)$$
$$(1\ \ 3\ \ 2… n\text{-}2\ \ n\ \ n\text{-}1) \cong (1\ \ n\ \ n\text{-}2 … 2\ \ 3\ \ n\text{-}1)$$
$$\vdots$$

Step3/ Make the process cycle for all each permutation resulted in the step 2 after deleting the equivalents to generate *n*!/2 permutation in this step as following:-

$$(1\ \ 2\ \ 3… n\text{-}2\ \ n\text{-}1\ \ n)$$
$$(2\ \ 3\ \ 4… n\text{-}1\ \ n\ \ 1)$$
$$(3\ \ 4\ \ 5… n\ \ 1\ \ 2)$$
$$(4\ \ 5\ \ 6… 1\ \ 2\ \ 3)$$
$$\vdots$$
$$(n\ \ 1\ \ 2… n\text{-}3\ \ n\text{-}2\ \ n\text{-}1)$$

$$(1\ \ 2\ \ 3… n\text{-}2\ \ n\ \ n\text{-}1)$$
$$(2\ \ 3\ \ 4\ …\ n\ \ n\text{-}1\ \ 1)$$
$$(3\ \ 4\ \ 5\ …\ n\text{-}1\ \ 1\ \ 2)$$
$$(4\ \ 5\ \ 6\ …\ 1\ \ 2\ \ 3)$$
$$\vdots$$
$$(n\text{-}1\ \ 1\ \ 2\ …\ n\text{-}3\ \ n\text{-}2\ \ n)$$

$$(1\ \ 2\ \ 3… n\text{-}3\ \ n\ \ n\text{-}1)$$
$$(2\ \ 3\ \ 4\ …\ n\ \ n\text{-}1\ \ 1)$$
$$(3\ \ 4\ \ 5\ …\ n\text{-}1\ \ 1\ \ 2)$$
$$(4\ \ 5\ \ 6\ …\ 1\ \ 2\ \ 3)$$
$$\vdots$$
$$(n\text{-}1\ \ 1\ \ 2… n\text{-}2\ \ n\text{-}3\ \ n)$$

$$(1\ \ 3\ \ 2… n\text{-}2\ \ n\ \ n\text{-}1)$$
$$(3\ \ 2\ \ 4\ …\ n\ \ n\text{-}1\ \ 1)$$
$$(2\ \ 4\ \ 5\ …\ n\text{-}1\ \ 1\ \ 3)$$
$$(4\ \ 5\ \ 6\ …\ 1\ \ 3\ \ 2)$$
$$\vdots$$
$$(n\text{-}1\ \ 1\ \ 3… n\text{-}3\ \ n\text{-}2\ \ n)$$

$$\vdots$$

Step4/ Take the inverse (reverse) for each permutations resulted from step 3 to produce all the required permutations of the permutation

$$
\begin{array}{ll}
(1 \quad 2 \quad 3\ldots n\text{-}2 \quad n\text{-}1 \quad n) & (n \quad n\text{-}1 \quad n\text{-}2\ldots 3 \quad 2 \quad 1) \\
(2 \quad 3 \quad 4 \ldots n\text{-}1 \quad n \quad 1) & (1 \quad n \quad n\text{-}1\ldots 4 \quad 3 \quad 2) \\
(3 \quad 4 \quad 5 \ldots n \quad 1 \quad 2) & (2 \quad 1 \quad n \ldots 5 \quad 4 \quad 3) \\
(4 \quad 5 \quad 6 \ldots 1 \quad 2 \quad 3) & (3 \quad 2 \quad 1 \ldots 6 \quad 5 \quad 4) \\
\qquad\qquad \vdots & \qquad\qquad \vdots \\
(n \quad 1 \quad 2\ldots n\text{-}3 \quad n\text{-}2 \quad n\text{-}1) & (n\text{-}1 \quad n\text{-}2 \quad n\text{-}3\ldots 2 \quad 1 \quad n)
\end{array}
$$

$$
\begin{array}{ll}
(1 \quad 2 \quad 3\ldots n\text{-}2 \quad n \quad n\text{-}1) & (n\text{-}1 \quad n \quad n\text{-}2\ldots 3 \quad 2 \quad 1) \\
(2 \quad 3 \quad 4 \ldots n \quad n\text{-}1 \quad 1) & (1 \quad n\text{-}1 \quad n \ldots 4 \quad 3 \quad 2) \\
(3 \quad 4 \quad 5 \ldots n\text{-}1 \quad 1 \quad 2) & (2 \quad 1 \quad n\text{-}1 \ldots 5 \quad 4 \quad 3) \\
(4 \quad 5 \quad 6 \ldots 1 \quad 2 \quad 3) & (3 \quad 2 \quad 1 \ldots 6 \quad 5 \quad 4) \\
\qquad\qquad \vdots & \qquad\qquad \vdots \\
(n\text{-}1 \quad 1 \quad 2 \ldots n\text{-}3 \quad n\text{-}2 \quad n) & (n \quad n\text{-}2 \quad n\text{-}3\ldots 2 \quad 1 \quad n\text{-}1)
\end{array}
$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

These are all steps for the perfect algorithm to generating permutation therefore to understand this algorithm we will give example to use this algorithm on initial permutation (1 2 3 4 5) that have 5 elements, then we most make these steps in the following

Step1/ Find the original starter sets by fixing two elements, and changes the remainder elements as following

$$
\begin{aligned}
(1\ 2\ 3\ 4\ 5) &\cong (1\ 4\ 3\ 2\ 5) \\
(1\ 2\ 4\ 3\ 5) &\cong (1\ 3\ 4\ 2\ 5) \\
(1\ 3\ 2\ 4\ 5) &\cong (1\ 4\ 2\ 3\ 5)
\end{aligned}
$$

Then delete the equivalents invers permutation, we will get original starter sets as following

$$
\begin{aligned}
&(1\ 2\ 3\ 4\ 5) \\
&(1\ 2\ 4\ 3\ 5) \\
&(1\ 3\ 2\ 4\ 5)
\end{aligned}
$$

Step2/ Find sequent starts sets by fixing another two elements for each permutation produced from step 1 as following

(1 2 3 4 5)	(1 2 4 3 5)	(1 3 2 4 5)
(1 2 3 5 4) ≅ (1 5 3 2 4)	(1 2 4 5 3) ≅ (1 5 4 2 3)	(1 3 4 5 2) ≅ (1 5 4 3 2)
(1 2 5 3 4) ≅ (1 3 5 2 4)	(1 2 5 4 3) ≅ (1 4 5 2 3)	(1 3 5 4 2) ≅ (1 4 5 3 2)
(1 3 2 5 4) ≅ (1 5 2 3 4)	(1 4 2 5 3) ≅ (1 5 2 4 3)	(1 4 5 3 2) ≅ (1 3 5 4 2)

Then take only one side of permutations and delete the equivalents as following

(1 2 3 4 5)	(1 2 4 3 5)	(1 3 2 4 5)
(1 2 3 5 4)	(1 2 4 5 3)	(1 3 4 5 2)
(1 2 5 3 4)	(1 2 5 4 3)	(1 3 5 4 2)
(1 3 2 5 4)	(1 4 2 5 3)	(1 4 5 3 2)

Step3/ Make cyclic for all permutation generating from step 2 to find half all permutations generate from 5! It is 5! /2 = 60 permutations as following

(1 2 3 4 5)	(1 2 3 5 4)	(1 2 5 3 4)	(1 3 2 5 4)
(2 3 4 5 1)	(2 3 5 4 1)	(2 5 3 4 1)	(3 2 5 4 1)
(3 4 5 1 2)	(3 5 4 1 2)	(5 3 4 1 2)	(2 5 4 1 3)
(4 5 1 2 3)	(5 4 1 2 3)	(3 4 1 2 5)	(5 4 1 3 2)
(5 1 2 3 4)	(4 1 2 3 5)	(4 1 2 5 3)	(4 1 3 2 5)

```
(1 2 4 3 5)    (1 2 4 5 3)    (1 2 5 4 3)    (1 4 2 5 3)
(2 4 3 5 1)    (2 4 5 3 1)    (2 5 4 3 1)    (4 2 5 3 1)
(4 3 5 1 2)    (4 5 3 1 2)    (5 4 3 1 2)    (2 5 3 1 4)
(3 5 1 2 4)    (5 3 1 2 4)    (4 3 1 2 5)    (5 3 1 4 2)
(5 1 2 4 3)    (3 1 2 4 5)    (3 1 2 5 4)    (3 1 4 2 5)

(1 3 2 4 5)    (1 3 4 5 2)    (1 3 5 4 2)    (1 4 5 3 2)
(3 2 4 5 1)    (3 4 5 2 1)    (3 5 4 2 1)    (4 5 3 2 1)
(2 4 5 1 3)    (4 5 2 1 3)    (5 4 2 1 3)    (5 3 2 1 4)
(4 5 1 3 2)    (5 2 1 3 4)    (4 2 1 3 5)    (3 2 1 4 5)
(5 1 3 2 4)    (2 1 3 4 5)    (2 1 3 5 4)    (2 1 4 5 3)
```

Step4/ Take inverse for all permutation produced from step 3 to find all permutation from (1 2 3 4 5) is 5! Permutations as following

Permutations	inverse		permutations	inverse
(1 2 3 4 5)	(5 4 3 2 1)		(1 2 3 5 4)	(4 5 3 2 1)
(2 3 4 5 1)	(1 5 4 3 2)		(2 3 5 4 1)	(1 4 5 3 2)
(3 4 5 1 2)	(2 1 5 4 3)		(3 5 4 1 2)	(2 1 4 5 3)
(4 5 1 2 3)	(3 2 1 5 4)		(5 4 1 2 3)	(3 2 1 4 5)
(5 1 2 3 4)	(4 3 2 1 5)		(4 1 2 3 5)	(5 3 2 1 4)
(1 2 5 3 4)	(4 3 5 2 1)		(1 3 2 5 4)	(4 5 2 3 1)
(2 5 3 4 1)	(1 4 3 5 2)		(3 2 5 4 1)	(1 4 5 2 3)
(5 3 4 1 2)	(2 1 4 3 5)		(2 5 4 1 3)	(3 1 4 5 2)
(3 4 1 2 5)	(5 2 1 4 3)		(5 4 1 3 2)	(2 3 1 4 5)
(4 1 2 5 3)	(3 5 2 1 4)		(4 1 3 2 5)	(5 2 3 1 4)

permutations	inverse		permutations	inverse
(1 2 4 3 5)	(5 3 4 2 1)		(1 2 4 5 3)	(3 5 4 2 1)
(2 4 3 5 1)	(1 5 3 4 2)		(2 4 5 3 1)	(1 3 5 4 2)
(4 3 5 1 2)	(2 1 5 3 4)		(4 5 3 1 2)	(2 1 3 5 4)
(3 5 1 2 4)	(4 2 1 5 3)		(5 3 1 2 4)	(4 2 1 3 5)
(5 1 2 4 3)	(3 4 2 1 5)		(3 1 2 4 5)	(5 4 2 1 3)
(1 2 5 4 3)	(3 4 5 2 1)		(1 4 2 5 3)	(3 5 2 4 1)
(2 5 4 3 1)	(1 3 4 5 2)		(4 2 5 3 1)	(1 3 5 2 4)
(5 4 3 1 2)	(2 1 3 4 5)		(2 5 3 1 4)	(4 1 3 5 2)
(4 3 1 2 5)	(5 2 1 3 4)		(5 3 1 4 2)	(2 4 1 3 5)
(3 1 2 5 4)	(4 5 2 1 3)		(3 1 4 2 5)	(5 2 4 1 3)

permutations	inverse		permutations	inverse
(1 3 2 4 5)	(5 4 2 3 1)		(1 3 4 5 2)	(2 5 4 3 1)
(3 2 4 5 1)	(1 5 4 2 3)		(3 4 5 2 1)	(1 2 5 4 3)
(2 4 5 1 3)	(3 1 5 4 2)		(4 5 2 1 3)	(3 1 2 5 4)
(4 5 1 3 2)	(2 3 1 5 4)		(5 2 1 3 4)	(4 3 1 2 5)
(5 1 3 2 4)	(4 2 3 1 5)		(2 1 3 4 5)	(5 4 3 1 2)
(1 3 5 4 2)	(2 4 5 3 1)		(1 4 5 3 2)	(2 3 5 4 1)
(3 5 4 2 1)	(1 2 4 5 3)		(4 5 3 2 1)	(1 2 3 5 4)
(5 4 2 1 3)	(3 1 2 4 5)		(5 3 2 1 4)	(4 1 2 3 5)
(4 2 1 3 5)	(5 3 1 2 4)		(3 2 1 4 5)	(5 4 1 2 3)
(2 1 3 5 4)	(4 5 3 1 2)		(2 1 4 5 3)	(3 5 4 1 2)

In this step we find 5! = 120 permutation for generating initial permutation (1 2 3 4 5) contained 5 elements.

Some Initial Results

Lemma 1. There exist $(n-2)!/2$ original starter sets each n element for listing $n!$ permutations.

Proof: From n elements there are $n!$ permutation. If we fixing two elements from these n elements, then product $(n-2)$ elements. Then from $(n-2)$ elements product $(n-2)!$ permutations. These are $(n-2)!$ Permutation to have the permutations and inverse. Divided the permutation to 2, we will get $(n-2)!/2$ permutations without invers which is original starter sets.

Theorem 2. There exist n cycles for each sequent starter sets of n elements

Proof: Let $(1, 2, 3, …, n)$ permutations sequent starter set with fixing 1 and n elements we have

$$
\begin{array}{cccccc}
1 & 2 & 3 & 4… & n \\
2 & 3 & 4 … & n & 1 \\
3 & 4 & … & n & 1 & 2 \\
… & … & · · · \\
… & … & · · · \\
… & … & · · · \\
n & 1 & 2 & … & n\text{-}2 & n\text{-}1
\end{array}
$$

which produce n cycle.

Lemma 3. There is $(n-1)!/2$ sequent starter sets from original starter sets for n elements.

Proof: We have $n!/n$ from list permutations, equal $n(n-1)!/(n)$. Then $(n).(n-1)!/(n) = (n-1)!$ permutation from n elements (permutation and invers), then $(n-1)!/2$ permutation without invers which is call sequent starter sets that is produced all $n!$ permutations.

Theorem 4. There exists $n!/2$ permutations from $(n-2)!/2$ original starter sets for n elements under cycle's operation.

Proof: we have $(n-2)!/2$ original starter sets (by lemma 1).

Then $(n-2)!$ All original starter sets and inverse original.

So $(n-2)!$ Permutations from $(n-2)$ elements

We have $n!$ all permutations and inverse from n elements, so if we take only permutations without invers we most divide by 2

Then $n!/2$ permutations from $(n-2)!/2$ original starter sets.

Theorem 5. There is $(n-2)!/2$ sequent starter sets with fixing two elements product from each original starter set permutation.

Theorem 6. The generation of all distinct permutation can be obtained by $(n-2)!/2$ original starter sets.

Conclusions

The perfect algorithm to generating all $n!$ permutation based on generating original and sequent starter sets with fixing two elements is proposed. This algorithm is very easy, in addition to technique by fixing two elements which reduces the computation there is recursive circular step for completing all list $n!$ permutation. This study focuses only on the development of the algorithm. Moreover the practical implementation in processor of this algorithm should also be established so that the computation can be done faster.

References

[1].   A.King, (2006). Generating indecomposable permutations. Discrete Mathematics, 306, 508-518.
[2].   A. H. Mahdi, & A. S. Hadi, (2011). Encrypted Block Code. Journal of Engineering and Development, 15, 101-107.
[3].   D. E. Knuth, (1998). The Art of Computer Programming, Combinatorial Properties of Permutations, 3, pp.11–72.
[4].   F. Ives, F. (1976). Permutation enumeration: four new permutation algorithms. Comm. ACM, 68-72.
[5].   F. Ruskey, & Effler, S. (2003). A CAT algorithm for generating permutations with a fixed number of inversions. Information Processing Letters. Computer Science Web.com, 86, 107-112.
[6].   F. Ruskey, & B. Bauslaugh, (1990). Generating alternating permutations lexicographically, BIT 30, 17-26.

[7]. F. Ruskey, & G. Pruesse, (1994). Generating linear extensions fast. SIAM. J. Comput, 23, 373-386.

[8]. G. S. Akl, (1980). A new algorithm for generating derangements. BIT 20, 2-7.

[9]. G. Langdon, (1967). An Algorithm for generating permutations. Communication of ACM, 298-299.

[10]. H. Ibrahim, Z. Omar, & A. M. Rohni, (2010). New algorithm for listing all permutations. Modern Sciences, 89-94.

[11]. H. Ibrahim, L. M. Zake, & (2013). A Simple and Efficient Algorithm for Listing Permutation via Starter Set Method. Malaysian Journal of Fundamental and Applied Sciences, 191-195.

[12]. Johnson, S. M. (1963). Generation of permutations by Adjacent Transpositions. Math. Comput., 282-285.

[13]. L.M. Zake, (2011). Comparisons for Determinants of Special Matrices by Algorithm Proposed. Journal of Information and Computing Science, 049-054.

[14]. M. Iyer, M. (1995). Permutation generation using matrices. Retrieved January 2009, from http://www.ddj.com/184409671.

[15]. R. Dominique, & B. van, (1999). A multistack method for the fast generation of permutations with minimal length increasing subsequences. Information Processing Letters, vol. 69, no. 3, pp. 123 –126.

[16]. R. Sedgewick, (1977). Permutation generation methods. Computing surveys, 137-164.

[17]. S. Effler, & F. Ruskey, (2003). A CAT algorithm for generating permutations with a fixed number of inversions. Information Processing Letters, 86, 107-112.

[18]. S. Habeeb, (2012). Proposal for Complex AES Security using Key Generator and Text Permutation. Eng. & Tech. Journal, Vol.30, No.12, pp. 2067- 2075

[19]. T. Khanit, (2007). A computerized algorithm for generating permutation and its application in determining a determinant. World Academy of Science, Engineering and Technology, 27, 178-183.