PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

New Recursive Circular Algorithm for Listing All Permutations

Sharmila Karim[1]*, Zurni Omar[1], Haslinda Ibrahim[1], Khairil Iskandar Othman[2] and Mohamed Suleiman[3]

[1]College of Art and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
[2]Mathematics Department, Faculty of Information Technology and Quantitative Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[3]Mathematics Department, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ABSTRACT

Linear array of permutations is hard to be factorised. However, by using a starter set, the process of listing the permutations becomes easy. Once the starter sets are obtained, the circular and reverse of circular operations are easily employed to produce distinct permutations from each starter set. However, a problem arises when the equivalence starter sets generate similar permutations and, therefore, willneed to be discarded. In this paper, a new recursive strategy is proposed to generate starter sets that will not incur equivalence by circular operation. Computational advantages are presented that compare the results obtained by the new algorithm with those obtained using two other existing methods. The result indicates that the new algorithm is faster than the other two in time execution.

Keywords: Algorithm, circular, permutation, starter sets

INTRODUCTION

The generation of all $n!$ permutations of n elements is a fundamental problem in combinatorics and important in computing. Various methods on listing all permutations have been published and can be classified into two categories: (i) exchange-based techniques; (ii) non-exchange-based techniques (Sedgewick, 1977). The exchange-based techniques generate new permutations by making possible changes among two consecutive elements such as transposition of non-adjacent elements (Well, 1961; Heap, 1963), and transposition with adjacent elements (Trotter, 1962; Johnson, 1963; Ives, 1976; Gao & Wang, 2003; Viktorov, 2007; Borisenko *et al*. 2008). Whilst non-exchange-based techniques generate new permutations

Article history:
Received: 7 April 2011
Accepted: 6 August 2013

E-mail addresses:
mila@uum.edu.my (Sharmila Karim),
zurni@uum.edu.my (Zurni Omar),
linda@uum.edu.my (Haslinda Ibrahim),
khairil17@tmsk.uitm.edu.my (Khairil Iskandar Othman),
mohameds@upm.edu.my (Mohamed Suleiman)
*Corresponding Author

ISSN: 0128-7680

with certain restrictions such as lexicographic order (Ord-Smith, 1970), nested cyclic (Langdon, 1967), and partial reversion (Zaks, 1984; Shin, 2002; Thongchiew, 2007).

According to Sedgewick (1977), generating permutation under cycling restrictions is simpler and more powerful compared to other restriction techniques. Langdon (1967) and Iyer (1995) proposed a cycling technique where the main idea is to start with cycling interchange of n elements until two elements are cycled. However, Iyer's (1995) technique is only valid for $n < 5$ because repetition of permutation occurs when $n > 4$.

In spite of that, Ibrahim *et al*. (2010) introduced a new permutation technique based on distinct starter sets that employ circular and reversing operations. The crucial task of these operations is to generate the distinct starter sets by eliminating the equivalence starter sets. Although this technique is simple and easy to use, unfortunately, eliminating the equivalence starters is quite tedious when the number of elements increases. This paper attempts to overcome this drawback by introducing a new strategy for generating distinct starter sets without eliminating the equivalence starter sets.

MATERIALS AND METHODS

Preliminary definition

The following definitions will be used throughout this paper.

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

Definition 2. An equivalence starter set is a set that can produce the same permutation from other starter sets.

Definition 3. The reverse set is a set that is produced by reversing the order of the permutation set.

Definition 4. A Latin square of order n is an $n \times n$ array in which n distinct symbols are arranged where each symbol occurs once in each row and column.

Definition 5. The circular permutation (CP) of order n is a Latin square of order n.

Definition 6. The reverse of circular permutation (RoCP) is also a Latin square of order n which is obtained by reversing arrangement elements in each row of circular permutation.

Example 1. Consider $n = 4$ and the fixed element is 1. There are two starters: (1234) and (1432).

The circular process is applied on both starters. The CP of each starter is listed as followed:

1234	1432
2341	4321
3412	3214
4123	2143

We may then apply the reversing process to either CP of the starter sets e.g. (1234) and RoCP as follows:

4321
1432
2143
3214

The RoCP of the starter (1234) generates the same permutation CP of (1432). Therefore, we refer to (1432) as the equivalence starter set of (1234). That equivalence starter set needs to be discarded. With a new algorithm, the equivalence starter sets will not be generated.

The development of the algorithm

The general algorithm for permutation generation follows:

Let S be the set of n elements i.e. $(1,2,3,\ldots,n-3,n-2,n-1,n)$

Step 1: Set $(1,2,3,4,\ldots,n-3,n-2,n-1,n)$ is taken as the initial permutation and it is assumed to be without loss of generality; therefore, the first element is fixed.

Step 2: Identify the last three elements of initial permutation from Step 1. By employing CP to the last three elements in initial permutation from step 1 three other distinct starter sets are produced, as shown below:

1, 2, …, n−3, n−2, n−1, n
1, 2, …, n−3, n−1, n, n−2
1, 2, …, n−3, n, n−2, n−1

Step 3: Identify the last four elements of each starter set in Step 2. By employing CP to the final four elements in each starter set in Step 2, 12 distinct starter sets are obtained, as shown below.

1, 2, …, *n*−3, *n*−2, *n*−1, *n*	**1, 2, …, *n*−3, *n*−1, *n*, *n*−2**	**1, 2, …, *n*−3, *n*, *n*−2, *n*−1**
1, 2,…, n−2, n−1, n, n−3	1, 2, …, n−1, n, n−2, n−3	1, 2, …, n, n−2, n−1, n−3
1, 2, …, n−1, n, n−3, n−2	1, 2, …, n, n−2, n−3, n−1	1, 2, …, n−2, n−1, n−3, n
1, 2, …, n, n−3, n−2, n−1	1, 2, …, n−2, n−3, n−1, n	1, 2, …, n−1, n−3, n, n−2

⋮

Step n−2: Identify the last n−1 elements of each starter sets in step n−3 By employing to the last n−1 elements on each starter set in step n−3, the $\frac{(n-1)!}{2}$ distinct starter sets are obtained.

Step n−1: Perform CP and RoCP simultaneously on all n elements of the $\frac{(n-1)!}{2}$ distinct starter set, and n! distinct permutations are obtained.

Step n: Display all n! permutations.

There are ($n-2$) steps needed to generate a starter set, after which, the CP and RoCP are employed on these starter sets to list down all n! distinct permutations.

To illustrate this algorithm, let's consider the set of five elements, i.e. S = (1,2,3,4,5).

Step 1: Take Set (1, 2, 3, 4, 5) as the initial permutation that appears without loss of generality, and the first element is fixed.

Step 2: Identify the last three elements of the initial permutation from Step 1. By employing CP to the last three elements in the initial permutation from Step 1 three other distinct starter sets are produced, as shown below:

1, 2, 3, 4, 5
1, 2, 4, 5, 3
1, 2, 5, 3, 4

Step 3: Identify the last four elements of each starter set in Step 2. By employing CP to the last four elements on each starter set in Step 2, 12 distinct starter sets are obtained, as shown below:

1, 2, 4, 5, 3	**1, 2, 5, 3, 4**	**1, 2, 3, 4, 5**
1, 4, 5, 3, 2	1, 5, 3, 4, 2	1, 3, 4, 5, 2
1, 5, 3, 2, 4	1, 3, 4, 2, 5	1, 4, 5, 2, 3
1, 3, 2, 4, 5	1, 4, 2, 5, 3	1, 5, 2, 3, 4

Step 4: Perform CP and RoCP simultaneously to all n elements of the 12 distinct starter sets and 5! distinct permutations are obtained (see Table 1) .

Step 5: Display 5! distinct permutations (see Table 1).

TABLE 1
The 5! Distinct Permutations

CP	RoCP	CP	RoCP
14532	23541	**14253**	35241
45321	12354	42531	13524
53214	41235	25314	41352
32145	54123	53142	24135
21453	35412	31425	52413
15324	42351	**12534**	43521
53241	14235	25341	14352
32415	51423	53412	21435
24153	35142	34125	52143

TABLE 1 (*continued*)

CP	RoCP	CP	RoCP
41532	23514	41253	35214
13245	54231	**13452**	25431
32451	15423	34521	12543
24513	31542	45213	31254
45132	23154	52134	43125
51324	42315	21345	54312
12453	35421	**14523**	32541
24531	13542	45231	13254
45312	21354	52314	41325
53124	42135	23145	54132
31245	54213	31452	25413
15342	24351	**15234**	43251
53421	12435	52341	14325
34215	51243	23415	51432
42153	35124	34152	25143
21534	43512	41523	32514
13425	52431	**12345**	54321
34251	51243	23451	15432
42513	31524	34512	21543
25134	43152	45123	32154
51342	24315	51234	43215

Remark 2: Permutations in **bold** represent the 12 starter sets for case $n = 5$.

RESULTS AND DISCUSSION

Some theoretical results

The following lemmas and theorem are produced from the recursive circular permutation generation method.

Lemma 1. $2n$ distinct permutations are produced by each distinct starter set.

Proof: Suppose we have a starter set of $A = (1,2,3\ \dots\ ,n-1,\ n)$ with n distinct elements. By using definition 5 where all the elements are cycled to the left, n distinct permutations are obtained, as given below:

1	2	…	$n-2$	$n-1$	n
2	…	$n-2$	$n-1$	n	1
$n-2$	$n-1$	n	1	2	…
$n-1$	n	1	2	…	$n-2$
n	1	2	…	$n-2$	$n-1$

Following then from definition 6 and reversing each row of CP produces other n distinct permutations, as given below:

n	$n-1$	$n-2$	…	2	1
1	n	$n-1$	$n-2$ …	2	
…	2	1	n	$n-1$	$n-2$
$n-2$	…	2	1	n	$n-1$
$n-1$	$n-2$	…	2	1	n

Thus, 2n distinct permutations are produced. □

Lemma 2. There are $\frac{(n-1)!}{2}$ distinct starter sets which are generated recursively for $n \geq 3$ under circular operation.

Proof: Let (1, 2, 3, ..., $n-3$, $n-2$, $n-1$, n) be taken as the initial starter for any $n \geq 3$. By employing CP to the last three elements, three distinct starters are produced, as shown below:

1	2	3	...	$n-3$	$n-2$	$n-1$	n	(*starter* 1)
1	2	3	...	$n-3$	$n-1$	n	$n-2$	(*starter* 2)
1	2	3	...	$n-3$	n	$n-2$	$n-1$	(*starter* 3)

Then for each previous starter set, the last four elements are selected, and by employing CP on these elements of the previous starter sets, four distinct starters are produced, as shown below:

From starter 1,

1	2	3	...	$n-1$	$n-1$	n	$n-3$
1	2	3	...	$n-1$	n	$n-3$	$n-2$
1	2	3	...	n	$n-3$	$n-2$	$n-1$
1	2	3	...	$n-3$	$n-2$	$n-1$	n

From starter 2,

1	2	3	...	$n-1$	n	$n-2$	$n-3$
1	2	3	...	n	$n-2$	$n-3$	$n-1$
1	2	3	...	$n-2$	$n-3$	$n-1$	n
1	2	3	...	$n-3$	$n-1$	n	$n-2$

From starter 3,

1	2	3	...	n	$n-2$	$n-1$	$n-3$
1	2	3	...	$n-2$	$n-1$	$n-3$	$n-1$
1	2	3	...	$n-1$	$n-3$	n	$n-1$
1	2	3	...	$n-3$	n	$n-2$	$n-1$

Thus, at this stage, the total starter sets are 3 × 4 = 12. The process will be continued until the last ($n-1$) element is selected.

3 last elements	3 starter sets
4 last elements	4 starter sets
5 last elements	5 starter sets
($n-2$) last elements	($n-2$) starter sets
($n-1$) last elements	($n-1$) starter sets

By product rule, the number of starter sets is

$$3\times 4\times 5\times \ldots \times n-2\times n-1 \tag{1}$$

$$=\frac{1}{2}\times 2\times 3\times 4\times 5\times \ldots \times n-2\times n-1 \tag{2}$$

$$=\frac{1}{2}(n-1)! \tag{3}$$

Remark 3: For case $n = 2$ is impossible since it has only one distinct starter set while $\frac{(2-1)!}{2}=\frac{1}{2}$.

Theorem 1. The generation of all $n!$ distinct permutations can be obtained by $\frac{(n-1)!}{2}$ distinct starter sets.

Proof: Lemma 2, there are $\frac{(n-1)!}{2}$ distinct starter sets for $n\geq 3$ while from lemma 1, $2n$ distinct permutations are obtained by employing the circular and reversing process on the starter sets.

Thus, $\frac{(n-1)!}{2}\times 2n = n!$ permutations are generated.

Analysis of time computation

In this section, this new algorithm is compared with Langdon's (1967) algorithm and Thongchiew's (2007) algorithm because these algorithms fall under the non-exchange restriction. The comparison over time computation between the new algorithm, Langdon's (1967) algorithm and Thongchiew's (2007) algorithm is given in Table 2. The results are given in milliseconds and all the algorithms are written in C language.

As can be observed from Table 2, the new algorithm is faster than Langdon's (1967) algorithm and Thongchiew's (2007) algorithm. At $n = 9$, the new algorithm runs the same as Langdon's (1967), but almost twice as fast as Thongchiew's(2007). In other words, Langdon's algorithm runs two times slower than the new algorithm, and Thongchiew's (2007) is the slowest among the three algorithms.

TABLE 2
Time Computation of Algorithms

n	New algorithm	Langdon's (1967) algorithm	Thongchiew's (2007) algorithm
8	0	0	0
9	15	15	63
10	109	171	687
11	983	2012	7488
12	12839	26520	90106
13	173270	365213	1672510
14	2590946	5423443	22448205
15	41885652	85173825	246139962

CONCLUSION

A new approach to listing n! permutations that is based on recursive circular generated starter sets is proposed in this paper. Furthermore, this recursive circular algorithm is efficient as the starter sets can be generated without eliminating the equivalence starter sets, thus reducing computation time.

REFERENCES

Barisenko, A. A., Kalashnikov, V. V., Kulik, I. A., & Goryachev, O. E. (2008). Generation of permutations based upon factorial numbers, *IEEE*, 57-61.

Gao, J., & Wang, D. (2003). *Permutation generation: Two new permutation algorithms*. Retrieved from http://arxiv.org/abs/cs/0306025

Heap, B. (1963). Permutation by interchanges. *Computer Journal*, 293-294.

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

Ives, F. (1976). Permutation enumeration: Four new permutation algorithms. *Comm. ACM* , 68-72.

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

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

Langdon, G. (1967). An algorithm for generating permutations. *Communication of ACM*, 298-299.

Ord-Smith, J. (1970). Generation of permutation sequences Part 1. *Computer Journal*, 152-155.

Sedgewick, R. (1977). Permutation generation methods. *Computing surveys*, 137-164.

Shin, D. (2002). The permutation algorithm for non-sparse matrix determinant in symbolic computation. *Proceeding on the 15th CISL winter workshop*, (pp. 42-54). Japan.

Thongchiew, K. (2007). A computerize algorithm for generating permutation and its application in determining a determinant. *Proc. of World Academy of Science, Engineering and Technology, 21*, 178-183.

Trotter, H. F. (1962). Algorithm 115. Perm. *Comm. ACM* , 434-435.

Viktorov, O. (2007). Permutation generation algorithm. *Asian Journal of Information Technology* , 956-957.

Wells, M. (1961). Generation of permutation by transposition. *Math comp*, 192-195.

Zaks, S. (1984). A new algorithm for generation of permutations. *BIT, 24*, 196-204.