Discrete Applied Mathematics 166 (2014) 164–169

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Lexicographic ranking and unranking of derangements in cycle notation

Kenji Mikawa [a,*], Ken Tanaka [b]

[a] Center for Academic Information Service, Niigata University, 8050 Ikarashi 2-no-cho, Nishi-ku, Niigata 950-2181, Japan

[b] Faculty of Science, Kanagawa University, 2946 Tsuchiya, Hiratsuka, Kanagawa 259-1293, Japan

ARTICLE INFO

Article history:
Received 25 June 2012
Received in revised form 4 September 2013
Accepted 1 October 2013
Available online 21 October 2013

Keywords:
Combinatorial problem
Algorithm
Derangements
Ranking
Unranking

ABSTRACT

We present lexicographic ranking and unranking algorithms for derangements expressed in cycle notation. These algorithms run in $O(n \log n)$ time, require $O(n)$ space, and use $O(n)$ arithmetic operations. Similar algorithms that require less than or equal to $O(n \log n)$ time with $O(n)$ space complexity have not previously been proposed.

1. Introduction

A derangement over n integers $[n] = \{0, 1, \ldots, n-1\}$ is defined as a permutation $\delta = \delta_0\delta_1 \cdots \delta_{n-1}$ of the integers with no fixed points, i.e., $\delta_i \neq i$ for all $i \in [n]$. The exercise of counting all derangements is a typical example of the inclusion–exclusion principle. Ranking and unranking algorithms are often used in performance evaluation of computer systems and algorithms, or test pattern generation. Applications, such as the optimal stopping problem [4], security protocols [7,8], and the network distribution problem [3,12], where each component is not mapped to itself must require derangements. The number of derangements is given by the recurrence relation,

$$d_n = \begin{cases} 0 & \text{if } n = 1, \\ 1 & \text{if } n = 2, \\ (n-1)\,(d_{n-1} + d_{n-2}) & \text{if } n \geq 3. \end{cases} \tag{1}$$

The ranking function for a set of permutations S is a bijection from S to $\{0, 1, \ldots, |S| - 1\}$; the unranking function is its inverse.

There are some well-known algorithms for ranking and unranking permutations over $[n]$. For lexicographic ranking and unranking, there are $O(n \log n)$ time algorithms that rely on modular arithmetic based on inversion tables and binary search with complete binary trees [5,2]. Myrvold and Ruskey were the first to propose linear time and space algorithms for this problem, but without lexicographic order [10]. The stringent condition that derangements have no fixed points complicates the development of ranking and unranking algorithms not only for lexicographic order, but also for some particular order. We are not aware of any algorithms that run in less than or equal to $O(n \log n)$ time and requiring only $O(n)$ space for any order of derangements.

* Corresponding author. Tel.: +81 252626113; fax: +81 252626113.
E-mail addresses: mikawa@cais.niigata-u.ac.jp (K. Mikawa), ktanaka@info.kanagawa-u.ac.jp (K. Tanaka).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.10.001

Recently, several papers have been published on enumerating all derangements in constant time to go from one derangement to the next [1,6,9]. By applying the data structure proposed in [9] to our algorithms, we expect to realize $O(n \log n)$ time with $O(n)$ space algorithms for lexicographic ranking and unranking, or linear time and space algorithms for some particular order. In this paper, we propose $O(n \log n)$ time with $O(n)$ space algorithms for lexicographic ranking and unranking of derangements in cycle notation. Traditionally, the time complexity of ranking and unranking is estimated on a computational model that can perform arithmetic operations in constant time. We discuss this problem using the same computational model in line with other published papers.

2. Cycle notation and its inversion

The lexicographic ranking for the set of permutations of size n is an order preserving ranking function p for this set, that is, π precedes lexicographically τ, if and only if $p(\pi) < p(\tau)$. The ranking function p for $\pi = \pi_0\pi_1 \cdots \pi_{n-1}$ is computed by

$$p(\pi) = \sum_{i=0}^{n-1} v_i \cdot (n - i - 1)! \tag{2}$$

with its inversion vector $v = v_0v_1 \cdots v_{n-1}$, where v_i is the number of entries π_j such that $i < j$ and $\pi_i > \pi_j$. Once the inversion vector has been computed, $p(\pi)$ is obtained in linear time from Eq. (2). Therefore, many researchers have struggled to reduce the time complexity of computing the inversion vector. Whereas naive implementations require $O(n^2)$ time to compute an inversion vector, the time complexity can be improved to $O(n \log n)$ time using a binary search tree or a merge sort [5,2].

The cycle notation is an intuitive notation for the order of a permutation that gives a mapping from $[n]$ to $[n]$ as a list of disjoint cycles. Stanley introduced the *standard representation* for this notation and some properties were detailed in [11]. A permutation π is decomposed into one or several disjoint cycles. For example, the following permutation is decomposed into three cycles, (04), (165), and (23).

$$\pi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 3 & 2 & 0 & 1 & 5 \end{pmatrix}. \tag{3}$$

The order of the elements in a cycle does not matter as long as the elements rotate their position while maintaining the order of the permutation. For example, the three cycles (165), (651), and (516) are equivalent. The order of the cycles does not matter either, since they are pairwise disjoint. According to the *standard representation*, but with some differences where our algorithm focuses on the smallest elements, the elements in each cycle of a permutation are arranged so that the smallest element is placed in the last position and write the permutation by listing the cycles in increasing order of their smallest elements. Following the standard representation, even without the parentheses, we can determine the unique cycle structure of a given permutation. The above permutation is written as 4065132.

We denote by $\mathfrak{C}([n])$ the set of all derangements over $[n]$ represented in cycle notation. The binary relationship '$a \prec b$' means that a is a lexicographic predecessor of b. This can be expressed by the recurrence form, $a \prec b$ if either $a_0 < b_0$ or $(a_0 = b_0) \cap (a_1 \cdots a_{n-1} \prec b_1 \cdots b_{n-1})$. For convenience, we expand the relationship to a *prefix isolation*, denoted by $a \prec_{|i} b$, and defined as $a_0 < b_0$ for $i = 0$ and $(a_0 \cdots a_{i-1} = b_0 \cdots b_{i-1}) \cap (a_i < b_i)$ for $i > 0$. The prefix isolation subdivides the set of predecessors of a given derangement into n partitions. Let $A = \{a \mid a \prec b\}$ and $A_i = \left\{a \mid a \prec_{|i} b\right\}$ for $a, b \in \mathfrak{C}([n])$.

Obviously, $A = A_0 \cup \cdots \cup A_{n-1}$ and $A_i \cap A_j = \emptyset$ for $i \neq j$, $i, j \in [n]$.

Given a derangement $\sigma \in \mathfrak{C}([n])$, the first element σ_0 must always be chosen from $[n] \setminus \{0\}$, since no derangements have any fixed points, *i.e.*, $\sigma_0 \neq 0$ and every cycle has at least two elements. The second element σ_1 must be in the state of either $\sigma_1 = 0$ if the first two elements $\sigma_0\sigma_1$ form a cycle, or $\sigma_1 \in [n] \setminus \{0, \sigma_0\}$ if the first cycle of σ remains open. After the first cycle has been fixed as $\sigma_0\sigma_1$, the remaining elements are used to form new derangements of length $n - 2$ into the suffix $\sigma_2\sigma_3 \cdots \sigma_{n-1}$. The number of these derangements is $(n - 1) \times d_{n-2}$. On the other hand, when the first cycle remains open, i.e., $\sigma_1 \neq 0$, it continues to form new derangements of length $n - 1$ into the suffix $\sigma_1\sigma_2 \cdots \sigma_{n-1}$. The number of these derangements is $(n - 1) \times d_{n-1}$. Thus, the total number of derangements in cycle notation is easily derived using the same recurrence formula as Eq. (1).

We observe that v is the inversion vector of a derangement if and only if v does not have two consecutive 0s. No inversions with respect to σ_i are found, i.e., $v_i = 0$, if σ_i is the last element of the cycle. This is obvious behavior, since σ_i is the smallest element between σ_i and σ_{n-1}, and is always placed in the last position of a cycle as the cycle terminator. Thus, both $v_{n-1} = 0$ and $v_{n-2} = 1$ hold. We show in Table 1 the lexicographic list of cycle notations and their corresponding derangements.

3. Ranking

We apply the same definition as the inversion vector to cycle notation and define the appropriate ranking function for derangements. The lexicographic rank of $\sigma \in \mathfrak{C}([n])$ can be thought of as the number of predecessors of σ on the lexicographic list of all elements belonging to $\mathfrak{C}([n])$. By the inductive definition of lexicographic order, $r(\sigma)$ can be expressed by the conceptual formula:

$$r(\sigma) = \sum_{i=0}^{n-1} \left| \left\{ \sigma' \in \mathfrak{C}([n]) | \sigma' \prec_{|i} \sigma \right\} \right|. \tag{4}$$