TCS-TR-A-11-54

TCS Technical Report

Counting Primitive Sorting Networks by πDDs

by

Jun Kawahara, Toshiki Saitoh, Ryo Yoshinaka and Shin-ichi Minato

Division of Computer Science

Report Series A

October 17, 2011

Hokkaido University
Graduate School of
Information Science and Technology

Email: minato@ist.hokudai.ac.jp

Phone: +81-011-706-7682
Fax: +81-011-706-7682

Counting Primitive Sorting Networks by πDDs

Jun Kawahara[1,2] Toshiki Saitoh[1,2] Ryo Yoshinaka[3]
Shin-ichi Minato[2,1]

[1] ERATO MINATO Project, Japan Science and Technology Agency, Japan
{jkawahara,t-saitoh}@erato.ist.hokudai.ac.jp
[2] Graduate School of Information Science and Technology, Hokkaido University, Japan
minato@ist.hokudai.ac.jp
[3] Graduate School of Informatics, Kyoto University, Japan
ry@i.kyoto-u.ac.jp

Abstract

A primitive sorting network is a fundamental computational model which is important both in mathematics and in engineering to operate permutations. In this paper, we propose an efficient method to count the number of ways to construct minimum primitive sorting networks. We developed πDD vector, a new data structure to represent and manipulate multisets of permutations, based on πDDs, which are recently proposed by Minato. We succeeded in calculating the number 2,752,596,959,306,389,652 for $n = 13$, which has not been known in the past, where n is the width of the primitive sorting network.

1 Introduction

Permutations and combinations are two basic concepts in elementary combinatorics and discrete mathematics [4]. Permutations appear in various problems such as sorting, ordering, matching, coding and many other real-life situations. Permutations are also important in group theory since they correspond to bijective functions and generate symmetric groups. The πDD, proposed by Minato [5], is a compact and canonical data structure for sets of permutations provided with a rich family of fundamental algebraic operations over them (e.g., union and intersection). πDD offers a convenient means to compute different sets of permutations of diverse properties by rather naively combining those primitive operations based on the algebraic characteristic of the target.

A typical situation where the πDD shows its performance is when we would like to obtain all permutations whose "distance" is at most k from the initial permutation. For example, the distance of two configurations of a Rubik's Cube$^{\text{TM}}$ can be defined to be the minimum number of moves that transform one into the other. A configuration of a Rubik's Cube can be seen as a permutation of stickers,

where the initial configuration is the identity, while a legal move on a Rubik's Cube is the multiplication of a special kind of permutations. Since πDD provides a method to calculate the product of two sets of permutations, one can easily compute all the configurations that can be reached by at most k moves from the initial configuration [5].

In general, we assume a finite set of "elementary permutations" and are interested in the number of ways to obtain a specific permutation using at most k multiplications of them. To give a general method to compute the number, this paper proposes a generalization of πDD, which we call *πDD vector*. The πDD vector is a data structure for multisets of permutations, just like *t*he ZDD vector [6] manipulates multisets of sets. We incrementally construct a πDD vector for $k = 1, 2, \ldots$ that contains a permutation π with multiplicity m iff there are m ways to represent π as a composition of at most k elementary permutations, which is performed by a polynomial number of basic operations of the πDD vector.

We demonstrate the advantage of our method through an algorithm that counts the number of *primitive sorting networks*, which is an n-input and n-output network to sort any sequence of integers (see Figure 1(a)). It was well studied by Knuth [3] and is often used in customized hardware of cryptographic systems and signal processing systems. It consists of n vertical lines (*lines*, for short) and a number of horizontal lines (*comparators*, for short) each of which connects two adjacent vertical lines. The role of a comparator between the ith line the $(i+1)$th line is to sort the ith number and the $(i+1)$th number of the input sequence. The primitive sorting network has to output the sorted sequence against an arbitrary input sequence. We would like to count the number of primitive sorting networks which have minimum comparators (called *minimum*).

For mathematical convenience, we consider a *ladder lottery* (or ghost leg), which is well-known as "amidakuji" in Japan, instead of a primitive sorting network (see Figure 1(b)). A horizontal line (*bar*, for short) of a ladder lottery has the role of swapping two numbers instead of sorting. A ladder lottery which has n lines receives the sequence $a(1), \ldots, a(n)$ as the input and outputs the sequence $1, \ldots, n$, where $\{a(1), \ldots, a(n)\} = \{1, \ldots, n\}$, which represents the permutation a obviously. We say a ladder lottery ℓ is *minimum* if there is no ladder lottery with less bars for the same permutation. Since it can be shown that a minimum primitive sorting network is equivalent to a minimum ladder lottery for the *reverse permutation* π_{rev}, which is defined by $\pi_{\mathrm{rev}}(i) = n - i + 1$ for each i, we set our goal to count the number of minimum ladder lotteries for the reverse permutation.

By $B(n)$ we denote our target number. Yamanaka et al. [9] developed an algorithm based on the reverse search [1] to enumerate all minimum ladder lotteries with n lines for the reverse permutation, by which they obtained $B(11) =$ 5,449,192,389,984. It took about 15 days to find the number $B(11)$ [10]. Their algorithm runs in O(1) time per an output ladder lottery. Matthew showed $B(12) =$ 2,894,710,651,370,536 [8].

We present a simple counting algorithm for this problem which uses a poly-

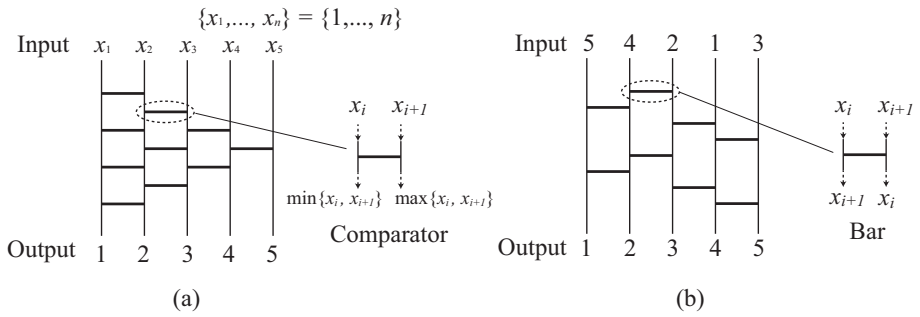


Figure 1: Examples of a primitive sorting network and a ladder lottery for $n = 5$.

nomial number of basic operations of the πDD vector, by which we can compute $B(12)$ in practical time. In addition, we improve the algorithm so that it computes the number of minimum ladder lotteries with n lines by the πDD vector for permutations over $\{1, \ldots, n-1\}$ by taking advantage of the special structure of minimum ladder lotteries. This technique enables us to compute $B(13) =$ 2,752,596,959,306,389,652, which has not been known in the past, in 473,314 seconds (about 5 days) on a 2.4GHz Opteron calculation server with 256GB memory.

2 Preliminaries

In this paper, we fix an integer $n \geq 2$.

2.1 Permutation

A *permutation* is a bijective function $\pi : S \to S$. This paper consider only permutations on $S = \{1, \ldots, n\}$. The application of π to an integer x is often denoted as $x\pi$ instead of $\pi(x)$. The product of two permutations is defined as their function composition: $x\pi_1\pi_2 = \pi_2(\pi_1(x))$. A permutation π is often specified by the sequence $(1\pi, \ldots, n\pi)$. The identical permutation $(1, 2, \ldots, n)$ is denoted by π_e. A *transposition* $\tau_{x,y}$ for $x, y \in S$ and $x \neq y$ is a permutation such that $x\tau_{x,y} = y$, $y\tau_{x,y} = x$, and $z\tau_{x,y} = z$ for $z \neq x, y$.

Proposition 1 (Minato [7])**.** *Any permutation π on S can be uniquely represented as a composition of transpositions $\pi = \tau_{x_1,y_1} \ldots \tau_{x_k,y_k}$ with $x_1, \ldots, x_k, y_1, \ldots, y_k \in S$ satisfying that $x_i > y_i$ for all $i = 1, \ldots, k$ and $x_i < x_{i+1}$ for all $i = 1, \ldots, k-1$.*

An *adjacent transposition* is a transposition $\tau_{x,y}$ with $y = x + 1$, which we abbreviate as τ_x. The inversion $\mathrm{inv}(\pi)$ of permutation π is defined by the number of pairs (i, j) such that $i, j \in S$, $i < j$ and $i\pi > j\pi$. For any permutation π and any integer i, either $\mathrm{inv}(\pi\tau_i) = \mathrm{inv}(\pi) + 1$ or $\mathrm{inv}(\pi\tau_i) = \mathrm{inv}(\pi) - 1$ holds because a pair $(i, i+1)$ is swapped and any other pair is not swapped by τ_i. A permutation and its inversion has the following well-known property.

Proposition 2. *For any permutation π, π is written as a product of* $\mathrm{inv}(\pi)$ *adjacent transpositions.*

2.2 πDD

Minato [7] has introduced a data structure for sets of permutations based on Proposition 1 and named it *πDD*. Here we give only a definition of the πDD. For more detailed explanations and implementation of the πDD, the reader is referred to Minato's original paper [7]. A πDD is defined to be a labeled directed acyclic graph satisfying the following properties.

- There are only two vertices **0** and **1** with outdegree 0, called the *0-terminal* and *1-terminal*.
- Each vertex except terminals has just 2 outgoing edges, which are labeled by 0 and 1 and called the *0-edge* and *1-edge*, respectively.
- Each vertex $\mathcal{P}$ except for the terminals is labeled by a pair of elements $(x_\mathcal{P}, y_\mathcal{P}) \in S \times S$ satisfying $x_\mathcal{P} > y_\mathcal{P}$.
- (*ordered*) If the 0-edge of a vertex $\mathcal{P}$ points $\mathcal{Q}$, then either $x_\mathcal{P} = x_\mathcal{Q}$ and $y_\mathcal{P} < y_\mathcal{Q}$ or $x_\mathcal{P} > x_\mathcal{Q}$ holds. If the 1-edge of a vertex $\mathcal{P}$ points $\mathcal{R}$, then $x_\mathcal{P} > x_\mathcal{R}$ holds.
- (*zero-suppression*) There is no vertex $\mathcal{P}$ whose 1-edge directly points the 0-terminal.
- (*uniqueness*) There are no distinct vertices $\mathcal{P}$ and $\mathcal{Q}$ such that
  - $(x_\mathcal{P}, y_\mathcal{P}) = (x_\mathcal{Q}, y_\mathcal{Q})$;
  - their 0-edges point the same vertex;
  - their 1-edges point the same vertex.

Each vertex $\mathcal{P}$ represents a set $\Pi_\mathcal{P}$ of permutations, which is recursively defined as follows:

- $\Pi_\mathbf{0} = \emptyset$ and $\Pi_\mathbf{1} = \{\pi_e\}$;
- $\Pi_\mathcal{P} = \Pi_\mathcal{Q} \cup (\Pi_\mathcal{R} \cdot \tau_{x_\mathcal{P}, y_\mathcal{P}}) = \Pi_\mathcal{Q} \cup \{\, \pi\tau_{x_\mathcal{P}, y_\mathcal{P}} \mid \pi \in \Pi_\mathcal{R} \,\}$ for a non-terminal vertex $\mathcal{P}$ whose 0-edge points $\mathcal{Q}$ and 1-edge points $\mathcal{R}$.

Minato [7] has proven that every set of permutation has a unique representation as a πDD, based on which, hereafter we simply write $\pi \in \mathcal{P}$ instead of $\pi \in \Pi_\mathcal{P}$ by identifying a vertex and the πDD consisting of the vertices reachable from that vertex. Minato presented some set operations on πDD, these operations are described in Table 1.

2.3 Multiset

In this section, we define notation for multisets. A multiset is a map on some set A into the set of the non-negative integers. A is called an underlying set of elements. For example,

$$\mathbf{P}(x) = \begin{cases} 3 & \text{if } x = a; \\ 1 & \text{if } x = b; \\ 0 & \text{if } x = c \text{ or } x = d \end{cases}$$

means a multiset which includes three a's and one b, where the underlying set is $\{a, b, c, d\}$. The number of x's in $\mathbf{P}$ is denoted by $\mathbf{P}.\,\mathrm{numberof}(x)$ or simply $\mathbf{P}(x)$. For two multisets $\mathbf{P}$ and $\mathbf{Q}$, we define the *multiset sum* by $(\mathbf{P} \uplus \mathbf{Q})(x) = \mathbf{P}(x) + \mathbf{Q}(x)$. For any multiset $\mathbf{P}$ whose underlying set is A and any set $B \subseteq A$, the subtraction B from $\mathbf{P}$ is defined by

$$(\mathbf{P} \setminus B)(x) = \begin{cases} 0 & \text{if } x \in B; \\ \mathbf{P}(x) & \text{otherwise.} \end{cases}$$

The indicator function of a set C is

$$\mathbf{1}_C(x) = \begin{cases} 1 & \text{if } x \in C; \\ 0 & \text{otherwise.} \end{cases}$$

The *zero multiset* is $\mathbf{0}(x) = 0$ for all x. We denote the set of elements that a multiset $\mathbf{P}$ contains by $\mathbf{P}.\,\mathrm{support} = \{x \,|\, \mathbf{P}(x) \geq 1\}$.

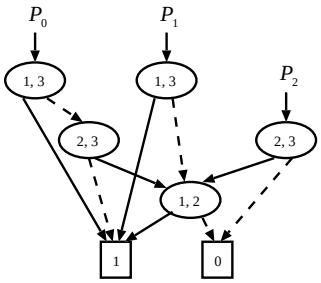
Let Π be all permutations over $\{1, \ldots, n\}$. For any permutation $\pi \in \Pi$ and any multiset $\mathbf{P}$ whose underlying set is Π, we define the product of $\mathbf{P}$ and π by

$$(\mathbf{P} \cdot \pi)(x) = \mathbf{P}(y) \text{ for all } x \in \Pi$$

where y is a permutation such that $x = y\pi$. This means to multiply each "element" in $\mathbf{P}$ by π.

2.4 Multiset of Permutations and πDDs

To represent and compute multisets of permutations efficiently, we introduce a *πDD vector* which is an array of πDDs. Let $\mathbf{P}$ be a multiset of permutations over $\{1, \ldots, n\}$. Let $m_{\mathbf{P}} = \max_\pi \mathbf{P}(\pi)$ and $k = \lfloor \log m_{\mathbf{P}} \rfloor$. Note that $\mathbf{P}(\pi)$ is the number of π's in $\mathbf{P}$. For any permutation π over $\{1, \ldots, n\}$, let $s_0^\pi, s_1^\pi, \ldots, s_k^\pi \in \{0, 1\}$ be such that $\mathbf{P}(\pi) = \sum_{0 \leq i \leq k} (2^i \times s_i^\pi)$, namely, the binary representation of $\mathbf{P}(\pi)$ is $(s_k^\pi \ldots s_1^\pi s_0^\pi)_2$. We define a *πDD vector $\vec{\mathcal{P}}$ which represents* $\mathbf{P}$ by $\vec{\mathcal{P}} = (\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_k)$ where all of $\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_k$ are πDDs such that for any permutation π and any index i, $\pi \in \mathcal{P}_i$ if and only if $s_i^\pi = 1$. We show an example of a πDD vector in Figure 2. In πDD, two vertices $\mathcal{P} = \mathcal{Q}$ if a set of permutations $\Pi_{\mathcal{P}}$ is equal to $\Pi_{\mathcal{Q}}$. We treat with an array of permutations so we handle a number of sets of permutations simultaneously in a monolithic memory space.



Permutation π	value
$(2,3,1)$	$\mathbf{P}(\pi) = 5$
$(3,2,1)$	$\mathbf{P}(\pi) = 3$
$(2,1,3)$	$\mathbf{P}(\pi) = 2$
$(1,2,3)$	$\mathbf{P}(\pi) = 1$

$\vec{\mathcal{P}} = (\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2)$.
$P_0 = \{(1,2,3), (2,3,1), (3,2,1)\}$
$P_1 = \{(2,1,3), (3,2,1)\}$
$P_2 = \{(2,3,1)\}$

Figure 2: πDD vector $\vec{\mathcal{P}}$.

Table 1: Primitive πDD operations.

$\mathcal{P}.top$	Returns IDs (x, y) at the root node of $\mathcal{P}$.
$\mathcal{P} \cup \mathcal{Q}$	Returns $\{\pi \mid \pi \in \mathcal{P}$ or $\pi \in \mathcal{Q}\}$.
$\mathcal{P} \cap \mathcal{Q}$	Returns $\{\pi \mid \pi \in \mathcal{P}$ and $\pi \in \mathcal{Q}\}$.
$\mathcal{P} \setminus \mathcal{Q}$	Returns $\{\pi \mid \pi \in \mathcal{P}$ and $\pi \notin \mathcal{Q}\}$.
$\mathcal{P}.\tau(x, y)$	Returns $\mathcal{P} \cdot \tau_{x,y}$.
$\mathcal{P} * \mathcal{Q}$	Returns $\{\alpha\beta \mid \alpha \in \mathcal{P}$ and $\beta \in \mathcal{Q}\}$.
$\mathcal{P}.cofact(x, y)$	Returns $\{\pi\tau_{x,y} \mid \pi \in \mathcal{P}$ and $x\pi = y\}$.
$\mathcal{P}.count$	Returns the number of permutations.

We introduce multiset operations on πDD vectors. Let $\mathbf{P}$ and $\mathbf{Q}$ be multisets of permutations, and $\vec{\mathcal{P}}$ and $\vec{\mathcal{Q}}$ be πDD vectors which represent $\mathbf{P}$ and $\mathbf{Q}$, respectively. We describe the multiset sum operation $\mathbf{P} \uplus \mathbf{Q}$ on πDD vectors as **Algorithm 1**. This algorithm implements a simple adder circuit for $\mathbf{P}$ and $\mathbf{Q}$. In this algorithm, πDD $\mathcal{R}_i$ represents the set of permutations π such that the ith digit s_i^π of $(\mathbf{P} \uplus \mathbf{Q})(\pi)$ is 1, and πDD $\mathcal{C}$ is the set of permutations that $\mathbf{P} \uplus \mathbf{Q}$ is carried up in the ith digit. By repeating this process, we obtain the πDD vector $\vec{\mathcal{R}} = (\mathcal{R}_0, \mathcal{R}_1 \ldots, \mathcal{R}_k)$ which represents $\mathbf{P} \uplus \mathbf{Q}$.

Algorithm 1 MULTISET SUM $\uplus$ ($\vec{\mathcal{P}} = (\mathcal{P}_0, \ldots, \mathcal{P}_{k_P})$, $\vec{\mathcal{Q}} = (\mathcal{Q}_0, \ldots, \mathcal{Q}_{k_Q})$)

$k \leftarrow \max(k_P, k_Q)$, $\mathcal{C} \leftarrow \emptyset$.
for $i = 0, 1, \ldots, k$ **do**
  $\mathcal{R}_i \leftarrow \mathcal{P}_i \oplus \mathcal{Q}_i \oplus \mathcal{C}$, $\mathcal{C} \leftarrow (\mathcal{P}_i \cap \mathcal{Q}_i) \cup (\mathcal{Q}_i \cap \mathcal{C}) \cup (\mathcal{P}_i \cap \mathcal{C})$.
end for
if $\mathcal{C} \neq \emptyset$ **then**
  $k \leftarrow k + 1$, $\mathcal{R}_k \leftarrow \mathcal{C}$
end if
return πDD vector $\vec{\mathcal{R}} = (\mathcal{R}_0, \mathcal{R}_1 \ldots, \mathcal{R}_k)$

Similarly, we can compute subtraction and multiplication on multisets of permutations. We describe some primitive πDD vector operations in Table 2. In this table, we consider that a multiset is identical to the corresponding πDD vector.

Table 2: Primitive πDD vector operations.

$\mathcal{A}.to\_multiset$	Returns $1_{\mathcal{A}}$.
$\vec{\mathcal{P}} \cdot \tau_{x,y}$	Returns $\mathbf{P'}$ s.t. $\mathbf{P'}(\pi \cdot \tau_{x,y}) = \mathbf{P}(\pi)$
$\vec{\mathcal{P}} \uplus \vec{\mathcal{Q}}$	Returns $\mathbf{P} \uplus \mathbf{Q}$.
$\vec{\mathcal{P}}.support$	Returns $\{\pi \mid \mathbf{P}(\pi) \geq 1\}$.
$\vec{\mathcal{P}} \setminus \mathcal{A}$	Returns $\mathbf{P} \setminus A$.
$\vec{\mathcal{P}}.numberof(\pi)$	Returns $\mathbf{P}(\pi)$.

3 Counting minimum ladder lotteries

3.1 Definition of a ladder lottery

In the paper, we treat only ladder lotteries with n lines. We define an alphabet $\Sigma_n = \{t_1, \ldots, t_{n-1}\}$. A ladder lottery (*ladder*, for short) ℓ which has n lines and k bars is a string $t_{a_1} \cdots t_{a_k}$ over Σ_n, where $a_i \in \{1, \ldots, n-1\}$ for each $i = 1, \ldots, k$. The ith character t_{a_i} of ladder $t_{a_1} \cdots t_{a_k}$ means that the ith bar connects the a_ith and the $(a_i + 1)$th lines.

For ladder $\ell = t_{a_1} \cdots t_{a_k}$, we write $\ell[i] = a_i$ and $\ell.\mathrm{sub}(i, j) = t_{a_i} t_{a_{i+1}} \cdots t_{a_j}$. We denote the length of ℓ by $|\ell|$. Let $L_n = \Sigma_n^*$ denote the set of all ladders with n lines. For $\ell = t_{a_1} \cdots t_{a_k} \in L_n$, we define $\mathrm{perm}(\ell) = \tau_{a_1} \cdots \tau_{a_k}$, which is called the *permutation represented by* ℓ. For any permutation π and any ladder $\ell \in L_n$ such that $\mathrm{perm}(\ell) = \pi$, if

$$\forall \ell' \in L_n,\ \mathrm{perm}(\ell') = \pi \Longrightarrow |\ell| \leq |\ell'|$$

is satisfied, then ℓ is a *minimum ladder for* π.

We now prove that each bar of a minimum ladder always swaps a pair (a, b) such that $a < b$ and increases the inversion of its permutation by one.

Lemma 3. *Let $k \leq n(n-1)/2$ be an integer. Then, $t_{a_1} \cdots t_{a_k}$ is a minimum ladder if and only if $\mathrm{inv}(\mathrm{perm}(t_{a_1} \cdots t_{a_i})) = i$ holds for all $i = 1, \ldots, k$.*

Proof. Suppose that $t_{a_1} \cdots t_{a_k}$ is minimum. Let $x_i = \mathrm{inv}(\mathrm{perm}(t_{a_1} \cdots t_{a_i}))$ for $i = 1, \ldots, k-1$. Trivially, $x_1 = \mathrm{inv}(\mathrm{perm}(t_{a_1})) = 1$ holds. Since for any permutation π and any adjacent transposition τ, either $\mathrm{inv}(\pi\tau) = \mathrm{inv}(\pi) + 1$ or $\mathrm{inv}(\pi\tau) = \mathrm{inv}(\pi) - 1$ is satisfied, $x_{i+1} - x_i = \pm 1$ holds for $i = 1, \ldots, k-1$. Therefore, $x_k = x_1 + \sum_{i=1}^{k-1}(x_{i+1} - x_i) \leq k$. Suppose that $x_k = h$ for some integer $h < k$. By Proposition 2, we can write $\mathrm{perm}(t_{a_1} \cdots t_{a_k}) = \tau_{b_1} \cdots \tau_{b_h}$ for some $b_1, \ldots, b_h$. Therefore, ladder $t_{b_1} \cdots t_{b_h}$ represents $\mathrm{perm}(t_{a_1} \cdots t_{a_k})$. However, this contradicts that $t_{a_1} \cdots t_{a_k}$ is minimum because $t_{b_1} \cdots t_{b_h}$ has just h bars. Hence, $x_k = k$. Since $x_{i+1} - x_i = \pm 1$ for all $i = 1, \ldots, k-1$ and $x_1 = 1$, we have $x_i = i$ for all $i = 1, \ldots, k$ and obtain the necessity.

For the sufficiency, suppose that $\mathrm{inv}(\mathrm{perm}(t_{a_1} \cdots t_{a_k})) = k$ and $t_{a_1} \cdots t_{a_k}$ is not minimum. Then, by the definition of a minimum ladder, there exists ladder

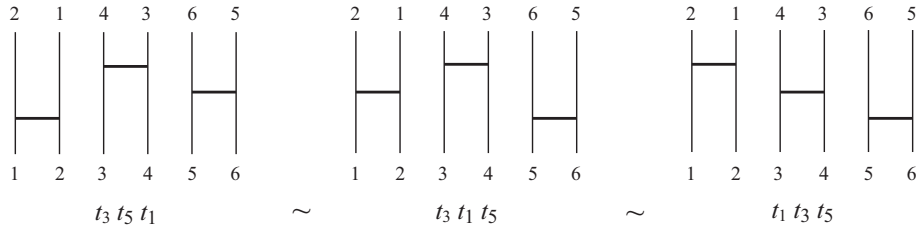


Figure 3: Ladder lotteries which belong to the same equivalence class. The rightmost one is normal.

$t_{c_1} \cdots t_{c_h}$ such that $\mathrm{perm}(t_{c_1} \cdots t_{c_h}) = \mathrm{perm}(t_{a_1} \cdots t_{a_k})$ for some $h < k$. By the same discussion as above, $\mathrm{inv}(\mathrm{perm}(t_{c_1} \cdots t_{c_h})) \leq h < k$. However, $\mathrm{inv}(\mathrm{perm}(t_{c_1} \cdots t_{c_h})) = \mathrm{inv}(\mathrm{perm}(t_{a_1} \cdots t_{a_k})) = k$. This is a contradiction. □

For $\ell = t_{a_1} \cdots t_{a_{i-1}} t_{a_i} t_{a_{i+1}} t_{a_{i+2}} \cdots t_{a_k} \in L_n$, we define $\mathrm{swap}(\ell, i) = t_{a_1} \cdots t_{a_{i-1}} t_{a_{i+1}} t_{a_i} t_{a_{i+2}} \cdots t_{a_k}$. If either $\ell[i+1] < \ell[i] - 1$ or $\ell[i+1] > \ell[i] + 1$ holds, then $\mathrm{perm}(\ell) = \mathrm{perm}(\mathrm{swap}(\ell, i))$, i.e., ℓ and $\mathrm{swap}(\ell, i)$ represents the same permutation (see Figure 3). For $\ell, \ell' \in L_n$, we define $\ell \sim \ell'$ iff there exist an integer $m \geq 1$, ladders $\ell_1, \ldots, \ell_m$, and integers $z_1, \ldots, z_{m-1}$ such that $\ell = \ell_1$, $\ell' = \ell_m$, $\ell_{i+1} = \mathrm{swap}(\ell_i, z_i)$ for $i = 1, \ldots, m-1$, and $(\ell_i[z_i + 1] < \ell_i[z_i] - 1$ or $\ell_i[z_i + 1] > \ell_i[z_i] + 1)$. Clearly, the relation $\sim$ is an equivalence relation of ℓ. We denote the equivalence class that contains ℓ by $C_n(\ell) = \{\ell' \in L_n \mid \ell \sim \ell'\}$.

We regard two ladders which belong to the same equivalence class as one ladder. To be accurate, for a given permutation π, the number of ladder for π should be defined as $|\{C_n(\ell) \mid \ell \in L_n, \mathrm{perm}(\ell) = \pi\}|$. To determine a representative of each equivalence class and count them, we introduce a normal form of a ladder.

Definition 4. A ladder $\ell \in L_n$ is in the *normal form* (simply, *normal*) if ℓ satisfies $\ell[i+1] \geq \ell[i] - 1$ for each $i = 1, \ldots, |\ell| - 1$.

We illustrate all normal minimum ladders for the reverse permutation for $n = 4$ in Figure 4. We now prove that there exists a bijection from the set of normal ladders to the set of equivalence classes C_n in Lemmas 5 and 7, namely, the number of equivalence classes C_n equals to that of normal ladders. Thus, we focus on counting distinct normal ladders.

Lemma 5. *For any ladder ℓ, there exists a normal ladder in $C_n(\ell)$.*

Proof. For $\ell' \in L_n$ and an integer m, let $P(\ell', m)$ be the proposition that $\ell'[i+1] \geq \ell'[i] - 1$ holds for each $i = 1, \ldots, m$. We now show the following statement: for a ladder ℓ', if ℓ' is not normal, $P(\ell', m)$ is satisfied and $P(\ell', m+1)$ is not satisfied, then there exists a ladder ℓ'' such that $\ell' \sim \ell''$ and $P(\ell'', m+1)$ are satisfied. Suppose that ℓ' has k bars. If $m = 0$ or $\ell'[m+2] < \ell'[i] - 1$ for all $i = 1, \ldots, m+1$, then $\ell'' = t_{\ell'[m+2]} t_{\ell'[1]} t_{\ell'[2]} \cdots t_{\ell'[m+1]} t_{\ell'[m+3]} t_{\ell'[m+4]} \cdots t_{\ell'[k]}$

satisfies $\ell' \sim \ell''$ and $P(\ell'', m+1)$. Otherwise, let j be an integer such that $\ell'[m+2] \geq \ell'[j] - 1$ and $\ell'[m+2] < \ell'[i] - 1$ for all $i = j+1, \ldots, m+1$. Then, $\ell'' = t_{\ell'[1]} \cdots t_{\ell'[j]} t_{\ell'[m+2]} t_{\ell'[j+1]} t_{\ell'[j+2]} \cdots t_{\ell'[m+1]} t_{\ell'[m+3]} t_{\ell'[m+4]} \cdots t_{\ell'[k]}$ satisfies $\ell' \sim \ell''$ and $P(\ell'', m+1)$. Therefore, the statement is obtained for both cases.

Note that for a ladder ℓ' with k bars, ℓ' is normal if $P(\ell', k-1)$ is satisfied. Applying the statement inductively, for any ladder ℓ, we can obtain the normal ladder ℓ' such that $\ell \sim \ell'$. □

Before we show uniqueness of the normal form of a ladder, we prove the following lemma.

Lemma 6. *For any integers i and j $(> i)$, and any normal ladder $\ell = t_{a_1} \cdots t_{a_k}$, suppose that $\ell[i] = y$ and $\ell[j] = x$. Then, all of $t_{x+1}, t_{x+2}, \ldots, t_{y-2}, t_{y-1}$ occur in $\ell.\mathit{sub}(i+1, j-1)$ if $x + 1 < y$.*

Proof. Suppose that at least one of $t_{x+1}, t_{x+2}, \ldots, t_{y-1}$ does not occur in $\ell.\mathrm{sub}(i+1, j-1)$. Let c be an integer such that t_c does not occur and all of $t_{c+1}, \ldots, t_{y-1}$ occur in $\ell.\mathrm{sub}(i+1, j-1)$. Let i' be the position where the last t_{c+1} occurs in $\ell.\mathrm{sub}(i+1, j-1)$. We now prove $a_{i''} > c$ for any $i'' \geq i' + 1$ by induction. By the property of a normal ladder, $a_{i'+1} \geq a_{i'} - 1 = (c+1) - 1 = c$. Since $a_{i'+1} \neq c$, $a_{i'+1} > c$. Therefore, this is true for $i' + 1$. Assume that $a_{i''} > c$. Then, $a_{i''+1} \geq a_{i''} - 1$. Since $a_{i''+1} \neq c$, $a_{i''+1} > c$ holds. Thus, $a_{i''} > c$ is satisfied for any $i'' \geq i' + 1$. However, since $i' < j$ and $a_j = x < c$, this is a contradiction. □

Lemma 7. *For any two distinct normal ladders $\ell, \ell' \in L_n$, $C_n(\ell) \neq C_n(\ell')$.*

Proof. Suppose that ℓ and ℓ' belong to the same equivalence class $C_n(\ell)$. Since ℓ and ℓ' are distinct, there exist $h \geq 0$ and $x, y \in \{1, \ldots, n-1\}$ such that

$$\ell = t_{a_1} t_{a_2} \cdots t_{a_h} t_x \cdots ,$$
$$\ell' = t_{a_1} t_{a_2} \cdots t_{a_h} t_y \cdots .$$

Without loss of generality, $x < y$. Since ℓ and ℓ' belong to the same equivalence class, there exist $\ell_1, \ldots, \ell_m \in C_n(\ell)$ and integers $z_1, \ldots, z_{m-1}$ such that $\ell_1 = \ell$, $\ell_m = \ell'$, $\ell_{i+1} = \mathrm{swap}(\ell_i, z_i)$ for all $i \in \{1, \ldots, m-1\}$, and $\ell_i \sim \ell_{i+1}$ for all $i \in \{1, \ldots, m-1\}$. Therefore, t_x must occur after the $(h+1)$th character in ℓ'. Let i be the position where the first t_x occurs after the hth character in ℓ'. By Lemma 6, t_{x+1} must occur in $\ell'.\mathrm{sub}(h+1, i-1)$. For a ladder ℓ'', let $\alpha(\ell'')$ be the position where the first t_x occurs and $\beta(\ell'')$ be the position where the first t_{x+1} occurs in $\ell''.\mathrm{sub}(h+1, |\ell''|)$. Then, let $P(\ell'')$ be the proposition $\alpha(\ell'') < \beta(\ell'')$. Clearly, $P(\ell)$ holds and $P(\ell')$ does not hold. Therefore, there exists j such that $P(\ell_j)$ holds and $P(\ell_{j+1})$ does not hold. Then, for some i', $\ell_j[i'] = x$, $\ell_j[i'+1] = x+1$, $\ell_{j+1}[i'] = x+1$ and $\ell_{j+1}[i'+1] = x$. Hence $\ell_j \not\sim \ell_{j+1}$. This is a contradiction. □

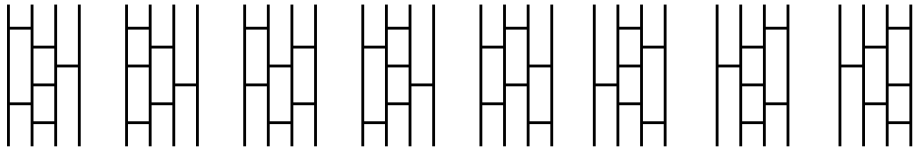


Figure 4: All normal minimum ladders for the reverse permutation when $n = 4$.

Let $S_i^{(k)}$ be the set of all ladders which has k bars and whose last character is t_i. For any normal ladder ℓ, if the $(j+1)$th character of ℓ is t_i, then the jth character of ℓ must be one of $t_1, \ldots, t_i$ or t_{i+1}. On the other hand, for any normal ladder s whose last character is t_i, all of $st_{i-1}, st_i, \ldots, st_{n-1}$ become normal ladders. Therefore, $S_i^{(k)}$ can be computed by the following recurrence relation:

$$S_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n; \\ \{t_i\} & \text{if } k = 1 \text{ and } 1 \leq i \leq n-1; \\ \{st_i \mid s \in \bigcup_{j=1}^{i+1} S_j^{(k-1)}\} & \text{otherwise.} \end{cases}$$

$\bigcup_{j=1}^{n-1} S_j^{(k)}$ is the set of all normal ladders which has k bars. Therefore, $\{\, \ell \in \bigcup_{j=1}^{n-1} S_j^{(k)} \mid \forall \ell' \in \bigcup_{i=1}^{k-1} \bigcup_{j=1}^{n-1} S_j^{(i)}, \mathrm{perm}(\ell) \neq \mathrm{perm}(\ell') \,\}$ is the set of normal minimum ladders whose inversion is k. Our basic idea to count the number of minimum ladder lotteries is based on this formula. However computing every set $S_j^{(k)}$ is too much for our purpose.

3.2 Counting minimum ladder lotteries using πDD

To count the number of minimum ladders for a given permutation π, it is enough to memorize $\mathrm{perm}(s)$ instead of $s \in S_j^{(k)}$ because $\mathrm{perm}(st_i) = \mathrm{perm}(s)\tau_i$ holds for any ladder s and any integer i. Therefore, we adopt the method by multisets of permutations described in the introduction. $\bar{\mathbf{P}}_i^{(k)}$ denotes the multiset of permutations which are represented by each element in $S_i^{(k)}$. This can be computed by the following recurrence relation:

$$\bar{\mathbf{P}}_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n; \\ \mathbf{1}_{\{\tau_i\}} & \text{if } k = 1 \text{ and } 1 \leq i \leq n-1; \\ (\biguplus_{j=1}^{i+1} \bar{\mathbf{P}}_j^{(k-1)}) \cdot \tau_i & \text{otherwise.} \end{cases}$$

We can efficiently compute $\bar{\mathbf{P}}_i^{(k)}$ using πDD vectors described in the previous section.

Lemma 8. *For any $i \in \{1, \ldots, n-1\}$, any integer k and any permutation π,*

$$\bar{\mathbf{P}}_i^{(k)}(\pi) = |\{s \mid s \in S_i^{(k)}, \mathit{perm}(s) = \pi\}|.$$

Proof. We use induction on k. When $k = 1$, the statement holds clearly. Suppose that $\bar{\mathbf{P}}_i^{(k-1)}(\pi) = |\{s \mid s \in S_i^{(k-1)}, \mathrm{perm}(s) = \pi\}|$ holds for any i when $k \geq 2$. Computing $\bar{\mathbf{P}}_i^{(k)}(\pi)$ according to the definition of the operations of multisets, we obtain

$$\begin{aligned}
\bar{\mathbf{P}}_i^{(k)}(\pi) &= ((\bar{\mathbf{P}}_1^{(k-1)} \uplus \cdots \uplus \bar{\mathbf{P}}_{i+1}^{(k-1)}) \cdot \tau_i)(\pi) \\
&= (\bar{\mathbf{P}}_1^{(k-1)} \cdot \tau_i)(\pi) + \cdots + (\bar{\mathbf{P}}_{i+1}^{(k-1)} \cdot \tau_i)(\pi) \\
&= \bar{\mathbf{P}}_1^{(k-1)}(\pi') + \cdots + \bar{\mathbf{P}}_{i+1}^{(k-1)}(\pi') \\
&= \sum_{j=1}^{i+1} |\{s \mid s \in S_j^{(k-1)}, \mathrm{perm}(s) = \pi'\}|,
\end{aligned}$$

where $\pi = \pi'\tau_i$.

Since the last character of any ladder $s \in S_j^{(k-1)}$ is t_j for any j, $S_j^{(k-1)} \cap S_{j'}^{(k-1)} = \emptyset$ holds if $j \neq j'$. Therefore,

$$\begin{aligned}
\sum_{j=1}^{i+1} |\{s \mid s \in S_j^{(k-1)}, \mathrm{perm}(s) = \pi'\}| &= |\{s \mid s \in \bigcup_{j=1}^{i+1} S_j^{(k-1)}, \mathrm{perm}(s) = \pi'\}| \\
&= |\{st_i \mid st_i \in S_i^{(k)}, \mathrm{perm}(s) = \pi'\}|.
\end{aligned}$$

Since $\mathrm{perm}(s) = \pi' \iff \mathrm{perm}(st_i) = \pi'\tau_i = \pi$, we have

$$\bar{\mathbf{P}}_i^{(k)}(\pi) = |\{s \mid s \in S_i^{(k)}, \mathrm{perm}(s) = \pi\}|.$$

□

$\bigcup_{i=1}^{n-1} \bar{\mathbf{P}}_i^{(k)}$ includes permutations which are represented by all normal ladders with k bars. We now show that in order to obtain only minimum normal ladders, we can remove all permutations which are not represented by minimum ladders from $\bigcup_{i=1}^{n-1} \bar{\mathbf{P}}_i^{(k)}$ for every k. We define a multiset $\mathbf{P}_i^{(k)}$ and a set $D^{(k)}$ by

$$\mathbf{P}_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n; \\ \mathbf{1}_{\{\tau_i\}} & \text{if } k = 1 \text{ and } 1 \leq i \leq n-1; \\ ((\biguplus_{j=1}^{i+1} \mathbf{P}_j^{(k-1)}) \cdot \tau_i) \setminus D^{(k-2)} & \text{otherwise,} \end{cases}$$

$$D^{(k)} = \begin{cases} \{\pi_e\} & \text{if } k = 0; \\ \bigcup_{i=1}^{n-1} (\mathbf{P}_i^{(k)}.\mathrm{support}) & \text{if } k \geq 1. \end{cases}$$

$D^{(k)}$ becomes the set of all permutations whose inversion is k.

Lemma 9. *For any $i \in \{1, \ldots, n-1\}$, any integer k and any permutation π,*

$$\mathbf{P}_i^{(k)}(\pi) = \begin{cases} |\{s \mid s \in S_i^{(k)}, \mathrm{perm}(s) = \pi\}| & \textit{if } \mathrm{inv}(\pi) = k; \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. For a multiset $\mathbf{P}$, we use notation $\Phi(\mathbf{P})$ instead of $\mathbf{P}.\mathrm{support}$. We now prove that $D^{(k)}$ equals to the set of all permutations whose inversion is k. We use induction on k. Since the permutation whose inversion is 0 is only π_e, $D^{(0)}$ is the set of all permutations whose inversion is 0. Suppose that $D^{(k-1)}$ is the set of all permutations whose inversion is $k-1$. Consider a permutation $\bar{\pi} \in D^{(k)} = \bigcup_{i=1}^{n-1} \Phi(\mathbf{P}_i^{(k)})$. For some i and j, $\bar{\pi} \in \Phi(\mathbf{P}_j^{(k-1)} \cdot \tau_i) \setminus D^{(k-2)}$. By the induction hypothesis, for any element π'' in $\Phi(\mathbf{P}_j^{(k-1)} \cdot \tau_i)$, either $\mathrm{inv}(\pi'') = k$ or $\mathrm{inv}(\pi'') = k-2$ holds. By the induction hypothesis, $D^{(k-2)}$ includes all permutations whose inversion is $k-2$. Therefore, the inversion of all permutations in $\Phi(\mathbf{P}_j^{(k-1)} \cdot \tau_i) \setminus D^{(k-2)}$ is k. Hence, $\mathrm{inv}(\bar{\pi}) = k$. Since we chose $\bar{\pi}$ arbitrarily, we can say that the inversion of any element in $D^{(k)}$ is k.

Consider a permutation π' such that $\mathrm{inv}(\pi') = k$. By Proposition 2, we can write $\pi' = \tau_{a_1} \cdots \tau_{a_k}$ for some $a_1, \ldots, a_k$. Since $\mathrm{inv}(\tau_{a_1} \cdots \tau_{a_{k-1}}) = k-1$ holds, $\tau_{a_1} \cdots \tau_{a_{k-1}} \in D^{(k-1)}$ is satisfied by the induction hypothesis. Then, $\tau_{a_1} \cdots \tau_{a_{k-1}} \in \Phi(\mathbf{P}_{a_{k-1}}^{(k-1)})$ holds. Therefore, $\pi' = \tau_{a_1} \cdots \tau_{a_k} \in \Phi(\mathbf{P}_{a_{k-1}}^{(k-1)} \cdot \tau_{a_k})$. Since $\mathrm{inv}(\pi') = k$ is also satisfied, $\pi' \notin D^{(k-2)}$ holds by the induction hypothesis. Therefore, $\pi' \in \mathbf{P}_{a_k}^{(k)}$ and $\pi' \in D^{(k)}$. We can say that all permutations whose inversion is k are in $D^{(k)}$. We obtain that $D^{(k)}$ equals to the set of all permutations whose inversion is k.

We use induction on k. When $k = 1$, the statement holds clearly. Suppose that

$$\mathbf{P}_i^{(k-1)}(\pi) = \begin{cases} |\{s \mid s \in S_i^{(k-1)}, \mathrm{perm}(s) = \pi\}| & \text{if } \mathrm{inv}(\pi) = k-1; \\ 0 & \text{otherwise.} \end{cases}$$

Computing $\mathbf{P}_i^{(k)}(\pi)$, we obtain

$$\begin{aligned} \mathbf{P}_i^{(k)}(\pi) &= (((\mathbf{P}_1^{(k-1)} \uplus \cdots \uplus \mathbf{P}_{i+1}^{(k-1)}) \cdot \tau_i) \setminus D^{(k-2)})(\pi) \\ &= \begin{cases} 0 & \text{if } \mathrm{inv}(\pi) = k-2 \\ (\mathbf{P}_1^{(k-1)} \cdot \tau_i)(\pi) + \cdots + (\mathbf{P}_{i+1}^{(k-1)} \cdot \tau_i)(\pi) & \text{otherwise} \end{cases} \\ &= \begin{cases} 0 & \text{if } \mathrm{inv}(\pi) = k-2 \\ \mathbf{P}_1^{(k-1)}(\pi') + \cdots + \mathbf{P}_{i+1}^{(k-1)}(\pi') & \text{otherwise,} \end{cases} \end{aligned}$$

where $\pi = \pi'\tau_i$. By the induction hypothesis,

$$\mathbf{P}_i^{(k)}(\pi) = \begin{cases} \sum_{j=1}^{i+1} |\{s \mid s \in S_j^{(k-1)}, \mathrm{perm}(s) = \pi'\}| & \text{if } \mathrm{inv}(\pi) \neq k-2 \\ & \quad \text{and } \mathrm{inv}(\pi') = k-1 \\ 0 & \text{otherwise.} \end{cases}$$

Since $\pi = \pi'\tau_i$ and $\mathrm{inv}(\pi) = \mathrm{inv}(\pi') \pm 1$,

$$\mathrm{inv}(\pi) \neq k-2 \text{ and } \mathrm{inv}(\pi') = k-1 \Longrightarrow \mathrm{inv}(\pi) = k.$$

Thus, we obtain

$$\mathbf{P}_i^{(k)}(\pi) = \begin{cases} \sum_{j=1}^{i+1} |\{s \mid s \in S_j^{(k-1)}, \mathrm{perm}(s) = \pi'\}| & \text{if } \mathrm{inv}(\pi) = k \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$\mathbf{P}_i^{(k)}(\pi) = \begin{cases} |\{s \mid s \in S_i^{(k)}, \mathrm{perm}(s) = \pi\}| & \text{if } \mathrm{inv}(\pi) = k; \\ 0 & \text{otherwise.} \end{cases}$$

□

Note that if $\mathrm{inv}(\pi) = k$, all ladders in $\{s \mid s \in S_i^{(k)}, \mathrm{perm}(s) = \pi\}$ are minimum for any i by Lemma 3. Therefore, if $\mathrm{inv}(\pi) = k$, $\mathbf{P}_i^{(k)}(\pi)$ gives the number of minimum ladders whose last character is t_i for π. We present an algorithm CountLadder to compute $\mathbf{P}_i^{(k)}$ and count minimum normal ladders.

Algorithm 2 CountLadder(n, π)

```
D_1 ← ∅, D_2 ← ∅
for i = 1, ..., n − 1 do
  P_i ← 1_{τ_i}
end for
P_n ← 0
for k = 2, ..., inv(π) do
  D ← 0, P_sum ← P_1
  for i = 1, ..., n − 1 do
    P_sum ← P_sum ⊎ P_{i+1}
    P_i ← (P_sum · τ_i) \ D_2
    D ← D ∪ (P_i.support)
  end for
  D_2 ← D_1, D_1 ← D
end for
return (⨄_{j=1}^{n−1} P_j).numberof(π)
```

Theorem 10. *For a given integer n and a permutation π,* CountLadder*(n, π) returns the number of minimum normal ladders for π with n lines correctly.*

Proof. After the end of the algorithm, $\mathbf{P}_i$ corresponds to $\mathbf{P}_i^{(\mathrm{inv}(\pi))}$. By Lemma 9, $(\biguplus_{j=1}^{n-1} \mathbf{P}_j)(\pi)$ gives the number of minimum normal ladders for π. □

3.3 Counting minimum ladder lotteries using diagonal bars

Consider any permutation π. For any minimum ladder $\ell = t_{a_1} \cdots t_{a_k}$, all of $t_1, \ldots, t_{1\pi^{-1}}$ must occur in ℓ in this order (see Figure 5(a)).

Lemma 11. *Every minimum ladder ℓ is uniquely represented as*

$$\ell = \alpha_0 t_1 \alpha_1 t_2 \alpha_2 \ldots t_{d-1} \alpha_{d-1},$$

where $d = 1(\mathrm{perm}(\ell))$, $\alpha_i \in \Sigma_n^*$ and neither t_i nor t_{i+1} occurs in α_i.

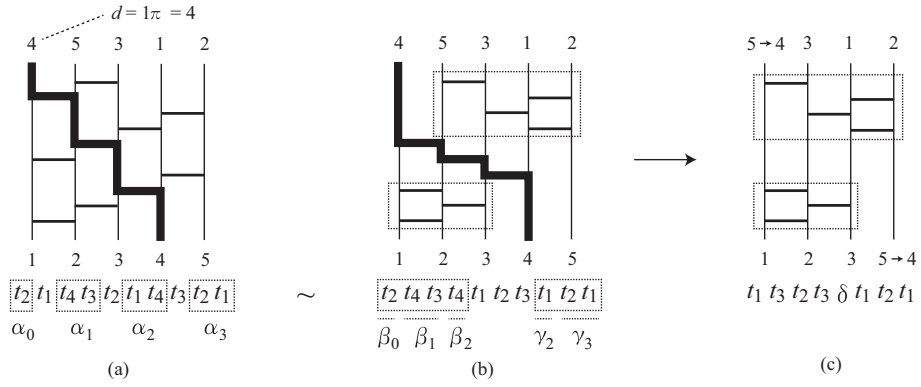


Figure 5: Diagonal bars and β-γ decomposition of a ladder.

Proof. All of $t_1, \ldots, t_{d-1}$ occur in ℓ because $d = 1(\mathrm{perm}(\ell))$. We define p_i by

$$p_i = \begin{cases} 0 & \text{if } i = 0; \\ \text{the position where the first } t_i \text{ occurs in } \ell & \text{if } i = 1; \\ \text{the position where the first } t_i \text{ occurs} & \\ \qquad \text{after the } p_{i-1}\text{th character in } \ell & \text{if } 2 \leq i \leq d-1; \\ |\ell| + 1 & \text{if } i = d. \end{cases}$$

Let $\ell_x = t_{b_1} t_{b_2} \cdots t_{b_x}$ for any integer $x \leq k$. We define $g(x) = 1(\mathrm{perm}(\ell_x))$, i.e., the number to which 1 is mapped by the permutation represented by the ladder which has the first x bars of ℓ.

We now prove the following statements (i) and (ii) are satisfied for $i = 1, \ldots, d-1$: (i) $g(p_i) = i + 1$ and (ii) t_i does not occur between the $(p_i + 1)$th character and the $(p_{i+1} - 1)$th character in ℓ.

First, we prove (i) for $i = 1$. By the definition of p_1, t_1 does not occur between the first character and the $(p_1 - 1)$th character in ℓ. Therefore, $g(1) = g(2) = \cdots = g(p_1 - 1) = 1$ and $g(p_1) = 2$. Hence, for $i = 1$, (i) holds.

Next, we prove that if (i) holds for some $i \in \{1, \ldots, d-1\}$, (ii) holds for i. Assume that (i) holds for some $i \in \{1, \ldots, d-1\}$. Suppose that t_i occurs between the $(p_i + 1)$th character and the $(p_{i+1} - 1)$th character in ℓ. Let q be the position where the first t_i occurs between the $(p_i + 1)$th character and the $(p_{i+1} - 1)$th character in ℓ. Since neither t_i nor t_{i+1} occurs between the $(p_i + 1)$th character and $(q-1)$th character in ℓ, $g(p_i) = g(p_i + 1) = \cdots = g(q-1) = i + 1$ hold by the assumption $g(p_i) = i+1$. Since the qth character of ℓ is t_i, $\mathrm{perm}(\ell_q) = \mathrm{perm}(\ell_{q-1})\tau_i$ holds. Let $\pi = \mathrm{perm}(\ell_{q-1})$, and a and b be the integers such that $a\pi = i$ and $b\pi = i + 1$. By the definition of g, $b = 1$ and $a > 1$. Therefore, multiplying π by τ_i decreases its inversion by 1. However, since ℓ is minimum, by Lemma 3, $\mathrm{inv}(\mathrm{perm}(\ell_{q-1})) = q-1$ and $\mathrm{inv}(\mathrm{perm}(\ell_q)) = q$. This is a contradiction. Therefore, t_i does not occur between the $(p_i + 1)$th character and the $(p_{i+1} - 1)$th character in ℓ.

Finally, we prove that if both (i) and (ii) hold for some $i \in \{1, \ldots, d-2\}$, (i) holds for $i+1$. Assume that both (i) and (ii) hold for some $i \in \{1, \ldots, d-2\}$. By this assumption and the definition of p_{i+1}, neither t_i nor t_{i+1} occurs between the (p_i+1)th character and the $(p_{i+1}-1)$th character. Therefore, $g(p_i) = g(p_i+1) = \cdots = g(p_{i+1}-1) = i+1$. Since the p_{i+1}th character is t_{i+1}, $g(p_{i+1}) = g(p_{i+1}-1)+1 = i+2$.

We proved that both (i) and (ii) are satisfied for $i = 1, \ldots, d-1$. Since (ii) holds for $i = 1, \ldots, d-1$, we obtain the statement of the lemma immediately. □

We consider a minimum ladder $\ell = \alpha_0 t_1 \alpha_1 t_2 \alpha_2 \ldots t_{d-1}\alpha_{d-1}$ where $\alpha_0, \ldots, \alpha_{d-1}$ satisfy the condition of Lemma 11. Let β_i be the string which is obtained by removing all of $t_1, \ldots, t_{i-1}$ from α_i, and γ_i be the string which is obtained by removing all of $t_{i+2}, \ldots, t_{n-1}$ from α_i. For any i, any j $(\leq i)$ and any j' $(\geq i+1)$, $\alpha_i \sim \beta_i\gamma_i$, $t_j\beta_i \sim \beta_i t_j$, $\gamma_i t_{j'} \sim t_{j'}\gamma_i$, and $\beta_i\gamma_j \sim \gamma_j\beta_i$ hold. Therefore,

$$\begin{aligned} \alpha_0 t_1 \alpha_1 t_2 \alpha_2 \ldots t_{d-1}\alpha_{d-1} &\sim \beta_0\gamma_0 t_1 \beta_1\gamma_1 t_2 \beta_2\gamma_2 t_3 \cdots t_{d-1}\beta_{d-1}\gamma_{d-1} \\ &\sim \beta_0\beta_1\cdots\beta_{d-1} t_1 t_2 \cdots t_{d-1}\gamma_0\gamma_1\cdots\gamma_{d-1}. \end{aligned}$$

This transformation is called *β-γ decomposition of ladder ℓ* (see Figure 5(b)). We call the bars $t_1 t_2 \cdots t_{d-1}$ in the middle of ℓ *diagonal bars*. Note that all of $\beta_0, \ldots, \beta_{d-1}, \gamma_0, \ldots, \gamma_{d-1}$ are uniquely determined by Lemma 11. We introduce another normal form of a ladder based on β-γ decomposition.

Definition 12. Let ℓ be a minimum ladder, $\pi = \mathrm{perm}(\ell)$ and $d = 1\pi$. Then, ℓ is a *diagonal normal form* (*d-normal*, for short) if $\ell = t_{a_1}\cdots t_{a_k} t_1 t_2 \cdots t_{d-1} t_{b_1}\cdots t_{b_{k'}}$ for some integers k and k' and the following conditions are satisfied: (i) for any $i \in \{1, \ldots, k\}$, $a_i \in \{2, \ldots, n-1\}$, (ii) the ladder $t_{a_1}\cdots t_{a_k}$ is normal, (iii) for any $i \in \{1, \ldots, k'\}$, $b_i \in \{1, \ldots, d-2\}$, and (iv) the ladder $t_{b_1}\cdots t_{b_{k'}}$ is normal.

We now prove that the number of equivalence classes C_n equals to that of d-normal ladders in Lemmas 13 and 14 similar to Lemmas 5 and 7. similar to Lemma 5.

Lemma 13. *For any ladder ℓ, there exists a d-normal ladder in $C_n(\ell)$.*

Proof. For any ladder ℓ, suppose that ℓ is decomposed into $\ell = \beta_0\beta_1\cdots\beta_{d-1} t_1 t_2 \cdots t_{d-1}\gamma_0\gamma_1\cdots\gamma_{d-1}$, where $d = 1(\mathrm{perm}(\ell))$. By Lemma 5, there exist normal ladders ℓ_β and ℓ_γ such that $\ell_\beta \sim \beta_0\beta_1\cdots\beta_{d-1}$ and $\ell_\gamma \sim \gamma_0\gamma_1\cdots\gamma_{d-1}$. The ladder $\ell_\beta t_1 t_2 \cdots t_{d-1}\ell_\gamma$ is d-normal and $\ell_\beta t_1 t_2 \cdots t_{d-1}\ell_\gamma \in C_n(\ell)$. □

Lemma 14. *For any two distinct d-normal ladders $\ell, \ell' \in L_n$, $C_n(\ell) \neq C_n(\ell')$.*

Proof. Suppose that $\ell \sim \ell'$, and let $\ell = \alpha_0 t_1 \alpha_1 t_2 \alpha_2 \cdots t_{d-1}\alpha_{d-1}$ and $\ell' = \alpha'_0 t_1 \alpha'_1 t_2 \alpha'_2 \cdots t_{d-1}\alpha'_{d-1}$, where $d = 1(\mathrm{perm}(\ell))$. Then, suppose that ℓ and ℓ' are decomposed into $\ell \sim \beta_0\beta_1\cdots\beta_{d-1} t_1 t_2 \cdots t_{d-1}\gamma_0\gamma_1\cdots\gamma_{d-1}$ and $\ell' \sim \beta'_0\beta'_1\cdots\beta'_{d-1} t_1 t_2 \cdots t_{d-1}\gamma'_0\gamma'_1\cdots\gamma'_{d-1}$, respectively.

We now prove $\beta_0\beta_1\cdots\beta_{d-1} \sim \beta'_0\beta'_1\cdots\beta'_{d-1}$ and $\gamma_0\gamma_1\cdots\gamma_{d-1} \sim \gamma'_0\gamma'_1\cdots\gamma'_{d-1}$. It is enough to show the case when $\ell' = \mathrm{swap}(\ell, z)$ for some z, namely, the following cases (i) and (ii): (i) $\alpha_i = \alpha'_i$ for $i \in \{1, 2, \ldots, p-1, p+1, \ldots, d-1\}$, $\alpha_p = \alpha'' t_x t_y \alpha'''$ and $\alpha'_p = \alpha'' t_y t_x \alpha'''$ for some strings α'' and α''' and some integers x and y, and (ii) $\alpha_i = \alpha'_i$ for $i \in \{1, 2, \ldots, p-2, p+1, \ldots, d-1\}$ and $\alpha'_{p-1} = \alpha_{p-1}t_x$ and $t_x\alpha'_p = \alpha_p$ for some integer $x \neq p-1$.

In the case of (i), this clearly holds because β-γ decompositions of ℓ and ℓ' are the same. In the case of (ii), both $\beta_i = \beta'_i$ and $\gamma_i = \gamma'_i$ hold for $i \in \{1, 2, \ldots, p-2, p+1, \ldots, d-1\}$. If $x \in \{1, \ldots, i-2\}$, then $\beta'_{p-1} = \beta_{p-1}$, $\beta'_p = \beta_p$, $\gamma'_{p-1} = \gamma_{p-1}t_x$ and $t_x\gamma'_p = \gamma_p$. Therefore, $\beta_0\beta_1\cdots\beta_{d-1} \sim \beta'_0\beta'_1\cdots\beta'_{d-1}$ and $\gamma_0\gamma_1\cdots\gamma_{d-1} \sim \gamma'_0\gamma'_1\cdots\gamma'_{d-1}$ hold. If $x \in \{i+1, \ldots, d-1\}$, then $\beta'_{p-1} = \beta_{p-1}t_x$, $t_x\beta'_p = \beta_p$, $\gamma'_{p-1} = \gamma_{p-1}$ and $\gamma'_p = \gamma_p$. Therefore, $\beta_0\beta_1\cdots\beta_{d-1} \sim \beta'_0\beta'_1\cdots\beta'_{d-1}$ and $\gamma_0\gamma_1\cdots\gamma_{d-1} \sim \gamma'_0\gamma'_1\cdots\gamma'_{d-1}$.

Hence, by Lemmas 5 and 7, for normal ladders ℓ_β and $\ell_{\beta'}$ such that $\ell_\beta \sim \beta_0\beta_1\cdots\beta_{d-1}$ and $\ell_{\beta'} \sim \beta'_0\beta'_1\cdots\beta'_{d-1}$, $\ell_\beta = \ell_{\beta'}$ holds, and for normal ladders ℓ_γ and $\ell_{\gamma'}$ such that $\ell_\gamma \sim \gamma_0\gamma_1\cdots\gamma_{d-1}$ and $\ell_{\gamma'} \sim \gamma'_0\gamma'_1\cdots\gamma'_{d-1}$, $\ell_\gamma = \ell_{\gamma'}$ holds. Therefore, for d-normal ladders ℓ'' and ℓ''' such that $\ell'' \sim \ell$ and $\ell''' \sim \ell'$, $\ell'' = \ell'''$ holds. □

We fix our target permutation π in the rest of this section. We will count minimum ladders for π. By Lemmas 13 and 14, it is enough to count all d-normal ladders. Let $d = 1\pi$. Consider a d-normal ladder $\ell = \ell_1 t_1 \cdots t_{d-1}\ell_2$ such that $\mathrm{perm}(\ell) = \pi$, where $\ell_1 = t_{a_1}\cdots t_{a_k}$ and $\ell_2 = t_{b_1}\cdots t_{b_{k'}}$. Note that $a_i \geq 2$ and $b_i \leq d-2$ for each i. We now construct a ladder $t_{a_1-1}t_{a_2-1}\cdots t_{a_k-1}t_{b_1}t_{b_2}\cdots t_{b_{k'}}$, which has $n-1$ lines and is obtained by removing diagonal bars from ℓ and shifting ℓ_1 to the left (see Figure 5(c)). To distinguish $\ell'_1 = t_{a_1-1}t_{a_2-1}\cdots t_{a_k-1}$ from ℓ_2, we insert the special character δ into $\ell'_1\ell_2$ between ℓ'_1 and ℓ_2 as $\ell'_1\delta\ell_2$, which we call an *r-ladder*. Note that ℓ'_1 is normal iff ℓ_1 is normal. Since a map $t_{a_1}\cdots t_{a_k}t_1\cdots t_{d-1}t_{b_1}\cdots t_{b_{k'}} \mapsto t_{a_1-1}t_{a_2-1}\cdots t_{a_k-1}\delta t_{b_1}t_{b_2}\cdots t_{b_{k'}}$ is a one-to-one correspondence, the number of all r-ladders is equal to the number of all minimum d-normal ladders.

We define the permutation ψ_π over $\{1, \ldots, n-1\}$ by

$$x\psi_\pi = \begin{cases} (x+1)\pi & \text{if } (x+1)\pi < d; \\ (x+1)\pi - 1 & \text{if } (x+1)\pi > d. \end{cases}$$

By a simple calculation, $\mathrm{perm}(t_{a_1-1}t_{a_2-1}\cdots t_{a_k-1}t_{b_1}t_{b_2}\cdots t_{b_{k'}}) = \psi_\pi$ if and only if $\mathrm{perm}(t_{a_1}\cdots t_{a_k}t_1\cdots t_{d-1}t_{b_1}\cdots t_{b_{k'}}) = \pi$. Therefore, the number of all r-ladders for ψ_π is equal to that of all minimum d-normal ladders for π.

We enumerate all r-ladders for ψ_π by the following recurrence relations:

$$S_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n-1; \\ \{t_i\} & \text{if } k = 1 \text{ and } 1 \le i \le n-2; \\ \{st_i \mid s \in \bigcup_{j=1}^{i+1} S_j^{(k-1)}\} & \text{otherwise,} \end{cases}$$

$$\hat{S}_i^{(k)} = \{s\delta \mid s \in S_i^{(k)}\},$$

$$V_i^{(k)} = \begin{cases} \emptyset & \text{if } i \ge d-1; \\ \{\delta t_i\} & \text{if } k = 1 \text{ and } 1 \le i \le d-2; \\ \{s\delta t_i \mid s \in \bigcup_{j=1}^{n-2} S_j^{(k-1)}\} & \\ \qquad \cup \{st_i \mid s \in \bigcup_{j=1}^{i+1} V_j^{(k-1)}\} & \text{otherwise.} \end{cases}$$

We introduce multisets of permutations similar to CountLadder.

$$\mathbf{P}_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n-1; \\ \mathbf{1}_{\{\tau_i\}} & \text{if } k = 1 \text{ and } 1 \le i \le n-2; \\ ((\biguplus_{j=1}^{i+1} \mathbf{P}_j^{(k-1)}) \cdot \tau_i) \setminus D^{(k-2)} & \text{otherwise,} \end{cases}$$

$$\mathbf{Q}_i^{(k)} = \begin{cases} \emptyset & \text{if } i = n-1; \\ \mathbf{1}_{\{\tau_i\}} & \text{if } k = 1 \text{ and } 1 \le i \le n-2; \\ (((\biguplus_{j=1}^{n-2} \mathbf{P}_j^{(k-1)}) & \\ \qquad \uplus (\biguplus_{j=1}^{i+1} \mathbf{Q}_j^{(k-1)})) \cdot \tau_i) \setminus D^{(k-2)} & \text{otherwise,} \end{cases}$$

$$D^{(k)} = \begin{cases} \{\pi_e\} & \text{if } k = 0; \\ \bigcup_{i=1}^{n-2} (\mathbf{P}_i^{(k)}.\mathrm{support}) & \text{if } k \ge 1. \end{cases}$$

Lemma 15. *For any $i \in \{1, \ldots, n-2\}$, any integer k and any permutation $\bar{\pi}$ over $\{1, \ldots, n-1\}$,*

$$\mathbf{P}_i^{(k)}(\bar{\pi}) = \begin{cases} |\{s \mid s \in \hat{S}_i^{(k)}, \mathrm{perm}(s) = \bar{\pi}\}| & \textit{if } \mathrm{inv}(\bar{\pi}) = k; \\ 0 & \textit{otherwise,} \end{cases} \tag{1}$$

$$\mathbf{Q}_i^{(k)}(\bar{\pi}) = \begin{cases} |\{s \mid s \in V_i^{(k)}, \mathrm{perm}(s) = \bar{\pi}\}| & \textit{if } \mathrm{inv}(\bar{\pi}) = k; \\ 0 & \textit{otherwise.} \end{cases} \tag{2}$$

Proof. We can obtain the statement similar to Lemma 9. □

By the above discussion, $((\biguplus_{j=1}^{n-2} \mathbf{P}_j^{(k)}) \uplus (\biguplus_{j=1}^{n-2} \mathbf{Q}_j^{(k)}))(\psi_\pi)$ is the number of minimum d-normal ladders for π, where $k = \mathrm{inv}(\psi_\pi) = \mathrm{inv}(\pi) - (d-1)$. We present an algorithm DiagCountLadder to compute $\mathbf{P}_i^{(k)}, \mathbf{Q}_i^{(k)}$ and count minimum ladders.

Theorem 16. *For a given integer n and a permutation π,* DiagCountLadder*(n, π) returns the number of minimum d-normal ladders for π with n lines correctly.*

Algorithm 3 DiagCountLadder(n, π)

$d \leftarrow 1\pi$
$D_1 \leftarrow \emptyset$, $D_2 \leftarrow \emptyset$
for $i = 1, \ldots, n-2$ **do**
  $\mathbf{P}_i \leftarrow \mathbf{1}_{\{\tau_i\}}$, $\mathbf{Q}_i \leftarrow \mathbf{1}_{\{\tau_i\}}$
end for
$\mathbf{P}_n \leftarrow \mathbf{0}$, $\mathbf{Q}_n \leftarrow \mathbf{0}$
for $k = 2, \ldots, \mathrm{inv}(\pi) - d + 1$ **do**
  $D \leftarrow 0$, $\mathbf{P}_{\mathrm{sum}} \leftarrow \mathbf{P}_1$
  for $i = 1, \ldots, n-2$ **do**
    $\mathbf{P}_{\mathrm{sum}} \leftarrow \mathbf{P}_{\mathrm{sum}} \uplus \mathbf{P}_{i+1}$
    $\mathbf{P}_i \leftarrow (\mathbf{P}_{\mathrm{sum}} \cdot \tau_i) \setminus D_2$
    $D \leftarrow D \cup (\mathbf{P}_i.\,\mathrm{support})$
  end for
  $\mathbf{P}_{\mathrm{sum}} \leftarrow \mathbf{P}_{\mathrm{sum}} \uplus \mathbf{Q}_1$
  for $i = 1, \ldots, n-2$ **do**
    $\mathbf{P}_{\mathrm{sum}} \leftarrow \mathbf{P}_{\mathrm{sum}} \uplus \mathbf{Q}_{i+1}$
    $\mathbf{Q}_i \leftarrow (\mathbf{P}_{\mathrm{sum}} \cdot \tau_i) \setminus D_2$
  end for
  $D_2 \leftarrow D_1, D_1 \leftarrow D$
end for
return $((\biguplus_{j=1}^{n-2} \mathbf{P}_j) \uplus (\biguplus_{j=1}^{n-2} \mathbf{Q}_j)).\,\mathrm{numberof}(\psi_\pi)$

Proof. After the end of the algorithm, $\mathbf{P}_i$ and $\mathbf{Q}_i$ corresponds to $\mathbf{P}_i^{(k)}$ and $\mathbf{Q}_i^{(k)}$ respectively, where $k = \mathrm{inv}(\pi) - d + 1$. By Lemma 15, $((\biguplus_{j=1}^{n-2} \mathbf{P}_j) \uplus (\biguplus_{j=1}^{n-2} \mathbf{Q}_j))(\psi_\pi)$ gives the number of minimum d-normal ladders for π. □

4 Experiments

We implemented the πDD vector manipulation system and added it to our πDD library. Using the enhanced library, we implemented CountLadder and DiagCountLadder algorithms described in the above sections by C++. We can describe each manipulation of multisets of permutations in CountLadder and DiagCountLadder directly. The source code of CountLadder is about 130 lines and that of DiagCountLadder is about 150 lines except for the library code.

Table 3 shows the number of minimum ladder lotteries for the reverse permutation π_{rev} with n lines, i.e., primitive sorting networks which have an n-input and n-output, and the running times of Yamanaka et al.'s algorithm [9, 10], CountLadder and DiagCountLadder. Their algorithm is implemented on a 3.0GHz Xeon, 8GB memory, MacOS X computer, while our algorithms are performed by a 2.4GHz Opteron calculation server with 256GB memory. For $n = 13$, the

Table 3: The number of minimum ladders for the reverse permutation (or minimum primitive sorting networks) and the counting time by Yamanaka et al.'s algorithm [10], CountLadder and DiagCountLadder.

n	Bars	Ladders	Algorithm of [10] Time (sec.)	CountLadder Time (sec.)	DiagCountLadder Time (sec.)
1	0	1	-	-	-
2	1	1	0	0.0	-
3	3	2	0	0.0	0.0
4	6	8	0	0.0	0.0
5	10	62	0	0.0	0.0
6	15	908	0	0.0	0.0
7	21	24698	0	0.0	0.1
8	28	1232944	0	1.0	0.3
9	36	112018190	25	14.6	4.1
10	45	18410581880	4230	232.1	62.0
11	55	5449192389984	1350409	4369.6	964.1
12	66	2894710651370536	-	109387.0	20172.6
13	78	2752596959306389652	-	-	473314.0

computation time of DiagCountLadder is 473,314 seconds (about 5 days). DiagCountLadder is 1400 times as fast as Yamanaka et al.'s algorithm for $n = 11$, but note that this is not quite a fair comparison because theirs enumerates all ladders, while ours counts them.

Table 4 shows the number of πDD nodes for $\mathbf{P}_i$ and $\mathbf{Q}_i$ $(i = 1, \ldots, n-1)$ when each loop in DiagCountLadder ends and the computation time in each loop in DiagCountLadder for $n = 13$. The maximum number of πDD nodes is 431,719,320 for $k = 37$.

5 Concluding Remarks

We presented algorithms based on multisets of permutations using a πDD vector to count ladder lotteries. Our method may be applied to various problems. Bulteau et. al. [2] showed that the sorting by transpositions problem, which is to compute the minimum number of "transpositions" to transform some permutation into the identity, is NP-hard. Their definition of "transposition" is to exchange a factor with indices $i, \ldots, j-1$ and $j, \ldots, k-1$. Our method for a ladder lottery can be extended for the sorting by transpositions directly.

References

[1] Avis, D. and Fukuda, K.: Reverse search for enumeration. Discrete Appl. Math. vol.65, no.1-3, 21–46 (1996)

Table 4: The number of πDD nodes and the computation time of the kth loop in DiagCountLadder when $n = 13$.

k	πDD size	time (sec)	k	πDD size	time (sec)	k	πDD size	time (sec)
2	84	0.0	24	74092105	1725.6	46	206952253	18962.3
3	412	0.0	25	97396108	2942.6	47	172465477	10645.3
4	993	0.0	26	124801718	3637.7	48	141373142	8651.3
5	2402	0.0	27	155963403	5758.8	49	112658584	13737.7
6	5131	0.0	28	190315087	12256.1	50	88033407	5515.3
7	11021	0.0	29	226782709	8714.8	51	66683750	4030.0
8	22118	0.1	30	264534765	10578.9	52	49297507	2873.2
9	43959	0.2	31	301462391	18219.5	53	35158712	2522.9
10	85782	0.4	32	337072133	14618.5	54	24319326	1456.8
11	164431	1.1	33	368133609	22549.8	55	16099126	1125.4
12	310427	2.3	34	395289444	24604.6	56	10210588	591.1
13	572021	4.8	35	414451469	26112.8	57	6127052	381.9
14	1028047	10.0	36	428085232	21489.7	58	3460221	213.6
15	1800491	19.4	37	431719320	28591.1	59	1827964	197.7
16	3063727	36.1	38	429673382	29019.7	60	886003	53.6
17	5061317	66.4	39	417344422	28869.4	61	389612	24.3
18	8118617	116.2	40	400535121	28683.0	62	150435	10.5
19	12621834	181.7	41	375016870	27763.5	63	49872	3.7
20	19045533	309.6	42	346931175	20068.8	64	12553	1.1
21	27883283	464.4	43	312915938	25127.3	65	2165	0.4
22	39668678	694.8	44	278670267	22752.8	66	6	0.0
23	54935314	1507.4	45	241865730	14816.2			

[2] Bulteau, L., Fertin, G., and Rusu, I.: Sorting by transpositions is difficult. In Proc. of the 38th International Colloquium on Automata, Languages and Programming (ICALP'11), vol.I, 654–665 (2011)

[3] Knuth, D.E.: Axioms and hulls. LNCS, vol. 606, 1–98 (1992)

[4] Knuth, D.E.: Combinatorial properties of permutations. The Art of Computer Programming, vol. 3, chap. 5.1, 11–72 (1998)

[5] Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In Proc. of 30th ACM/IEEE Design Automation Conference (DAC'93), 272–277 (1993)

[6] Minato, S.: VSOP (Valued-Sum-of-Products) Calculator for Knowledge Processing Based on Zero-Suppressed BDDs. Federation over the Web, LNAI, vol. 3847, 40-58 (2006).

[7] Minato, S.: PiDD: A New Decision Diagram for Efficient Problem Solving in Permutation Space. In Proc. of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT'11), LNCS, vol. 6695, 90–104 (2011)

[8] Sloane, N.: The On-Line Encyclopedia of Integer Sequences. `http://oeis.org/A006245` (accessed October 3, 2011)

[9] Yamanaka, K., Nakano, S., Matsui, Y., Uehara, R., and Nakada, K.: Efficient Enumeration of All Ladder Lotteries and Its Application. Theoretical Computer Science, vol.411, no.16-18, 1714–1722 (2010)

[10] Yamanaka, K.: Personal communication (2011).