Ranking and Unranking Restricted Permutations

Peter Kagey

November 1, 2022

Abstract

We discuss efficient methods for unranking derangements and ménage permutations. That is, given a list of these restricted permutations in lexicographic order, we will provide an algorithm to efficiently extract the k-th permutation from the list. We will show that this problem can be reduced to the problem of computing the number of restricted permutations with a given prefix, and then we will use rook theory to solve this counting problem. This has applications to combinatorics, probability, statistics, and modeling.

1 Overview and preliminaries

In January 2020, Richard Arratia sent out an email announcing a talk he was going to give about what came to be named "unranking derangements". Arratia noted that it is straightfoward to enumerate derangements (i.e. permutations without fixed points) in both senses of the word: both counting the number of derangements on n letters and listing them out one at a time. He asked, however, what if we want to find the element at a particular index somewhere in the middle of a lexicographically sorted list? Can one do this efficiently—in particular, can this be done without writing out the entire list? This suggests a broader combinatorial question: when is it possible to find the element at a particular index of a totally ordered set in some computationally efficient manner?

To be explicit about what we wish to compute efficiently, we define the notion of a *unranking*.

Definition 1. *Let $\mathscr{C}$ be a totally ordered finite set, and let $\{c_i\}_{i=1}^{|\mathscr{C}|}$ be the unique sequence of elements in $\mathscr{C}$ such that $c_i < c_{i+1}$ for all $1 \leq i < |\mathscr{C}|$. Then a* ***unranking*** *is a map*

$$\text{unrank}_{\mathscr{C}} \colon \{1, 2, \ldots, |\mathscr{C}|\} \to \mathscr{C}$$

that sends $i \mapsto c_i$.

An efficient way to unrank a collection of objects implies an efficient algorithm for sampling from the collection uniformly at random. This can potentially be of use in the case of Monte Carlo simulations and other instances where it is useful to be able to sample uniformly from a collection of combinatorial objects.

As the name suggests, every unranking problem comes with a dual ranking problem.

Definition 2. *A **ranking** of a totally ordered finite set $\mathscr{C}$ is a map*

$$\text{rank}_{\mathscr{C}} \colon \mathscr{C} \to \{1, 2, \ldots, |\mathscr{C}|\}$$

that sends $c_i \mapsto i$.

The existence of both efficient ranking and unranking maps implies the existence of an efficient encoding for these objects, which may be of interest to computer scientists. The encoding works by ranking an object to get its index, which then can be stored in as a positive integer and unranked on retrieval to recover the original object.

In January 2021, Richard Arratia announced a \$100 prize for developing an implementing an efficient algorithm for computing the unranking map in the context of *ménage permutations*. In particular, he stated the problem as follows:

Problem 3. *For $n = 20$ there are $A000179(20) = 312\,400\,218\,671\,253\,762 > 3.1 \cdot 10^{17}$ ménage permutations [4]. Determine the 10^{17}-th such permutation when listed in lexicographic order.*

In the remaining sections, we are show how to resolve all of the above problems in order to claim Richard Arratia's \$100 prize. That is, we will construct an algorithm for ranking and unranking both derangements and ménage permutations under the lexicographic ordering. We will show that the existence of an efficient way to count the number of such permutations with a given prefix implies that there is an efficient way to compute the ranking and unranking maps. Then we will develop some ideas from rook theory and apply them to the context of derangements and ménage permutations.

2 Prefix counting and word ranking

In both the case of unranking derangements and menage permutations (and in many other applications) our combinatorial objects are words in lexicographic order, which is a generalization of alphabetical order.

We begin by developing a general theory for unranking collections of words in lexicographic order by counting the number of words with a given prefix.

2.1 Words with a given prefix

We will start by introducing some basic definitions about words and prefixes, and to formalize the notion of lexicographic order.

Definition 4. *A finite* ***word*** *w over an alphabet $\mathscr{A}$ is a finite sequence $\{w_i \in \mathscr{A}\}_{i=1}^N$.*

The collection of finite words over the alphabet $\mathscr{A}$ is denoted by $\mathcal{W}_{\mathscr{A}}$, or just $\mathcal{W}$ when the alphabet is implicit from context.

Definition 5. *A word $w = \{w_i \in \mathscr{A}\}_{i=1}^N$ is said to begin with a* ***prefix*** *$\alpha = \{\alpha_i \in \mathscr{A}\}_{i=1}^M$ if $M \leq N$ and $w_i = \alpha_i$ for all $i \leq M$.*

Definition 6. *A word w is said to be before w' in* ***lexicographic order*** *if either w is a proper prefix of w', or if at the first position, i, where w and w' differ, $w_i < w'_i$.*

With these definitions established, we can turn the problem of unranking words into a problem about counting words with specified prefixes.

Theorem 7. *For $k > 0$, let $\mathcal{W}$ be a set of nonempty words on the alphabet $[n]$, and let $\mathscr{C} \subsetneq \mathcal{W}$ be a finite subset of words on this alphabet, with a total order given by its lexicographic order.*

Let $\#\operatorname{prefix}_{\mathscr{C}} \colon \mathcal{W} \to \mathscr{C}$ be the function that counts the number of words in $\mathscr{C}$ that begin with a given prefix.

Then the unranking function can be computed recursively by

$$\operatorname{unrank}_{\mathscr{C}}(i) = f_i^{\mathscr{C}}((1), 0) \tag{1}$$

where

$$f_i^{\mathscr{C}}(\alpha, j) = \begin{cases} f_i^{\mathscr{C}}(\alpha', j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)) & i > j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha) \quad \text{(2a)} \\ f_i^{\mathscr{C}}(\alpha'', j) & \alpha \notin \mathscr{C} \text{ and } i \leq j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha) \quad \text{(2b)} \\ f_i^{\mathscr{C}}(\alpha'', j+1) & \alpha \in \mathscr{C},\ i \leq j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha), \text{ and } i \neq j+1 \quad \text{(2c)} \\ \alpha & \alpha \in \mathscr{C} \text{ and } i = j+1, \quad \text{(2d)} \end{cases}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$, $\alpha' = (\alpha_1, \alpha_2, \ldots, \alpha_{\ell-1}, 1{+}\alpha_\ell)$, $\alpha'' = (\alpha_1, \alpha_2, \ldots, \alpha_\ell, 1)$, and j denotes the number of words in $\mathscr{C}$ that occur strictly before α.

Proof. We make four claims that we will prove using induction on the recursive applications of $f_i^{\mathscr{C}}(\alpha, j)$: that j is the number of words in $\mathscr{C}$ that occur strictly before α, that $\alpha \leq w_i$, and that the sequence of αs is strictly increasing.

This final claim (that the sequence of αs is strictly increasing) follows from the observation that $\alpha < \alpha'' < \alpha'$ in lexicographic order.

Because each iteration increases either j or ℓ or both, the number of recursive applications of $f_i^{\mathscr{C}}$ required to determine w_i is at most $i + \max_{w \in \mathcal{W}} |w|$, which is finite because $\mathcal{W}$ only contains of finite words.

The base case is clear: We start with $f_i^{\mathscr{C}}((1), 0)$ because (1) is the lexicographically earliest word, so 0 nonempty words strictly precede it, and $(1) \leq w_i$.

We will repeatedly use the observation that if j words precede α, then a word has prefix α if and only if its index is in $(j, j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)]$. Note that this range is empty whenever there are no words prefixed by α.

Case (2a). Because $j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)$ is the index of the last word that begins with α, if $i > j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)$, then w_i must begin with a length-ℓ prefix that is lexicographically later than α.

By construction, α' is the lexicographically earliest word of length ℓ that comes after α, therefore $\alpha' \leq w_i$. As such, the number of words that strictly precede α' is $j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)$, which is the sum of the number of words that occur strictly before α and the number of words that have prefix α.

Case (2b). If $\alpha \notin \mathscr{C}$ and $i \leq j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)$, then α is a *proper* prefix of w_i, and w_i is of length at least $\ell+1$. By construction, α'' is the lexicographically earliest word of length $\ell+1$ that has a prefix of α, so $\alpha'' \leq w_i$, and the number of words in $\mathscr{C}$ that precede α'' is equal to j, the number of words that precede α.

Case (2c). If $\alpha \in \mathscr{C}$, $i \leq j + \#\operatorname{prefix}_{\mathscr{C}}(\alpha)$, and $i \neq j+1$ then α must be the word at index $j + 1 < i$, because α itself is the lexicographically earliest word with the prefix α. Because words cannot appear multiple times in $\mathscr{C}$, w_i must have α as a *proper* prefix. Therefore $\alpha'' \leq w_i$ and the number of words that strictly precede α'' is $j + 1$: the number of words that strictly precede α plus α itself.

Case (2d). If $\alpha \in \mathscr{C}$ and $i = j + 1$, then $w_i = \alpha$ because α itself is the lexicographically earliest word with the prefix α, so it must occur at index $i = j + 1$. □

Notice that each recursive call of $f_i^{\mathscr{C}}$ increases the sum of the letters of α. If we suppose that $\mathscr{C}^{(\leq k)} \subsetneq \mathcal{W}_{[n]}$ is a finite set of words on the alphabet $[n]$ of length at most k, then unranking a word from $\mathscr{C}^{(\leq k)}$ requires at most nk recursive applications of $f_i^{\mathscr{C}^{(\leq k)}}$.

Therefore if there exists a polynomial time algorithm for computing $\#\operatorname{prefix}_{\mathscr{C}}$, then there exists an unranking algorithm that is polynomial in the size of the alphabet $\mathscr{A}$ and the length of the longest word. In the case of restricted permutations, each of these grow linearly with the number of letters in the ménage permutations.

2.2 Ranking words

Just as we can recursively find a word at a given index, we can also recursively find a index corresponding to a given word.

Claim 8. *If $\mathscr{C}$ is a collection of words such that no word is a prefix of another, then the rank of the word $w \in \mathscr{C}$ can be computed as the sum*

$$1 + \sum_{i=1}^{|w|} \sum_{i=1}^{w_i - 1} \#\operatorname{prefix}_{\mathscr{C}}(w_{(i)}^x)$$

where $w = (w_1, w_2, \ldots, w_{|w|})$ and $w_{(i)}^x = (w_1, w_2, \ldots, w_{i-1}, x)$.

3 Basic notions of rook theory

Now that we have shown that we can unrank words whenever we can compute the number of words with a given prefix, we want to develop techniques for this new counting problem. In the case of unranking derangements and permutations, it is useful to use ideas from rook theory, which provides a theory for understanding position-restricted permutations. Rook theory was introduced by Kaplansky and Riordan [1] in their 1946 paper *The Problem of the Rooks and its Applications*. In it, they discuss problems of restricted permutations in the language of rooks placed on a chessboard.

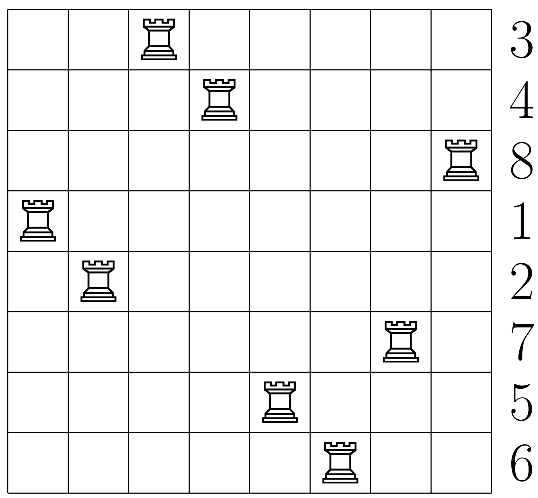


Figure 1: An illustration of the rook placement corresponding to the permutation $34812756 \in S_8$. A rook is placed in square $(i, \pi(i))$ for each i.

3.1 Definitions in rook theory

We begin by introducing some preliminary ideas from rook theory.

Definition 9. *A* ***board*** B *is a subset of* $[n] \times [n]$ *which represents the squares of an* $n \times n$ *chessboard that rooks are allowed to be placed on. Every board* B *has a complementary board* $B^c = ([n] \times [n]) \setminus B$*, which consists of all of the squares of* B *that a rook cannot be placed on.*

To each board, we can associate a generating polynomial that keeps track of the number of ways to place a given number of rooks on the squares of B in such a way that no two rooks are in the same row or column.

Definition 10. *The* ***rook polynomial*** *associated with a board* B*,*

$$p_B(x) = r_0 + r_1 x + r_2 x^2 + \cdots + r_n x^n, \tag{3}$$

is a generating polynomial where r_k denotes the number of k-element subsets of B such that no two elements share an x-coordinate or a y-coordinate.

In the context of permutations, we're typically interested in r_n, the number of ways to place n rooks on a restricted $n \times n$ board. However, it turns out that a naive application of the techniques from rook theory does not immediately allow us to count the number of restricted permutations with a given prefix. Computing the number of such permutations is known to be #P-hard for a board with arbitrary restrictions. We can see this by encoding a board B as a $(0,1)$-matrix and computing the matrix permanent, since there is a bijection between boards and $(0,1)$-matrices. (In fact, Shevelev [6] claims that "the theory of enumerating the permutations with restricted positions stimulated the development of the theory of the permanent.")

Lemma 11. *Let $M_B = \{a_{ij}\}$ be an $n \times n$ matrix where*

$$a_{ij} = \begin{cases} 1 & (i,j) \in B \\ 0 & (i,j) \notin B \end{cases}. \tag{4}$$

Then the coefficient of x^n in $p_B(x)$ is given by the matrix permanent

$$\text{perm}(M_B) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}. \tag{5}$$

Now is an appropriate time to recall Valiant's Theorem.

Theorem 12 (Valiant's Theorem [9])**.** *The counting problem of computing the permanent of a (0,1)-matrix is #P-complete.*

Corollary 13. *Computing the number of rook placements on an arbitrary $n \times n$ board is #P-hard.*

Therefore, in order to compute the number of permutations, we must exploit some additional structure of the restrictions.

3.2 Techniques of rook theory

Rook polynomials can be computed recursively. The base case is that for an empty board $B = \emptyset$, the corresponding rook polynomial is $p_\emptyset(x) = 1$, because there is one way to place no rooks, and no way to place one or more rooks.

Lemma 14 ([5])**.** *Given a board B and a square $(x, y) \in B$, we can define two resulting boards from including or excluding the given square:*

$$B_i = \{(x', y') \in B : x \neq x' \text{ and } y \neq y'\} \tag{6}$$
$$B_e = B \setminus (x, y). \tag{7}$$

Then we can write the rook polynomial for B in terms of this decomposition.

$$p_B(x) = x p_{B_i}(x) + p_{B_e}(x). \tag{8}$$

If we want to compute a rook polynomial using this construction, we can end up adding up lots of smaller rook polynomials—a number that is exponential in the size of B. When the number of squares that are missing from B is small, it can be easier to compute the rook polynomial of the complementary board, p_{B^c}, and use the principle of inclusion/exclusion on its coefficients to determine the rook polynomial for the original board, B.

In the case of derangements and ménage permutations, this is the strategy we'll use. We will start by finding the resulting board from a given prefix, find the rook polynomial of the complementary board, and use the principle of inclusion/exclusion to determine the number of ways to place rooks in the resulting board.

4 Unranking derangements

4.1 The answer to a $100 question about derangements

In January 2020, Richard Arratia sent out an email proposing a seminar talk. The title describes his first "$100 problem":

Problem 15. *"For* 100 *dollars, what is the* 500 *quadrillion-th derangement on* $n = 20$*?"*

Answer 16. *The author's computer program was able to compute the answer in less than twenty milliseconds. When written as words in lexicographic order, the derangement in* S_{20} *with rank* 5×10^{17} *is*

$$12\ 14\ 2\ 9\ 13\ 20\ 6\ 3\ 1\ 17\ 5\ 11\ 19\ 15\ 10\ 18\ 8\ 7\ 4\ 16. \tag{9}$$

4.2 Overview for unranking derangements

Arratia's question focused on unranking derangements written as words in lexicographic order. Other authors have looked at unranking derangements based on other total orderings. In particular, Mikawa and Tanaka [3] give an algorithm to rank/unrank derangements with respect to *lexicographic ordering in cycle notation.*

In this section we will develop an algorithm for ranking and unranking derangements with respect to their lexicographic ordering as words. The technique that we use will broadly be re-used in the next section. It is worthwhile to begin by recalling the definition of a derangement.

Definition 17. *A* ***derangement*** *is a permutation* $\pi \in S_n$ *such that* π *has no fixed points. That is, the set of derangements on* n *letters is*

$$\mathscr{D}_n = \{\pi \in S_n : \pi(i) \neq i \ \forall i \in [n]\}. \tag{10}$$

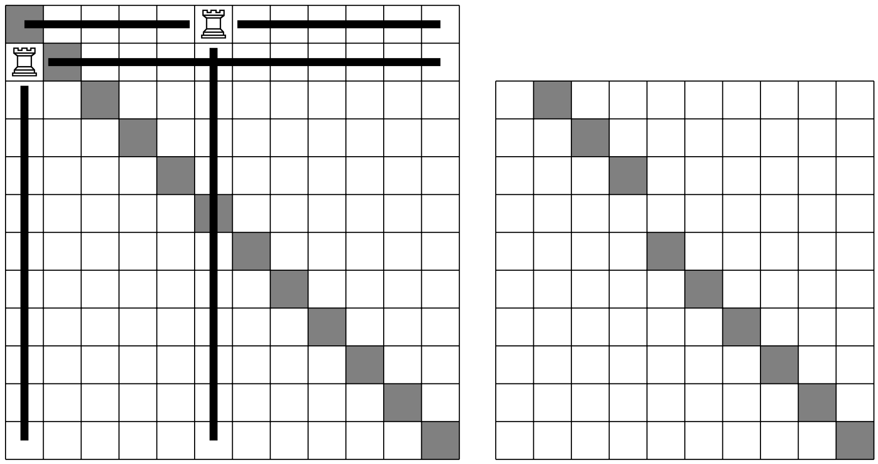


Figure 2: An example of a prefix $\alpha = (6, 1)$, and the board that results from deleting the first $\ell = 2$ rows and columns 6 and 1. The derived complementary board of B from α is $B^c_\alpha = \{(1, 2), (2, 3), (3, 4), (5, 5), \dots, (10, 10)\}$.

4.3 The complementary board

In order to compute the number of derangements with a given prefix, it is useful to look at the board that results after placing ℓ rooks according to these positions, as illustrated in Figure 2.

Definition 18. *If B is an $n \times n$ board, and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ is a valid prefix of length ℓ, the **derived board** of B from α, denoted B_α, is constructed by removing rows $1, 2, \dots, \ell$ and columns $\alpha_1, \alpha_2, \dots, \alpha_\ell$ from B, reindexing in such a way that both the row and column indexes are in $[n - \ell]$.*

*The **derived complementary board** B^c_α is the complement of B_α with respect to $[n - \ell] \times [n - \ell]$.*

Given a prefix of length ℓ, the number of ways of placing $n - \ell$ rooks on the derived board B_α is, by construction, equal to the number of words in $\mathscr{C}$ with prefix α

Lemma 19. *Given a valid ℓ-letter prefix $(\alpha_1, \alpha_2, \dots, \alpha_\ell)$ of a word on n letters, the number of squares in the derived complementary board is*

$$|B^c_\alpha| = n - \ell - |\{\ell + 1, \ell + 2, \dots, n\} \cap \{\alpha_1, \alpha_2, \dots, \alpha_\ell\}|, \tag{11}$$

and no two of these squares are in the same row or column.

Proof. Notice that the derived complementary board can be constructed in a different order: by first taking the complement, then deleting rows and columns, and finally reindexing the squares. Because the complementary board has no two squares in the same row or column, deleting and reindexing results in a derived complementary board with the same property.

Thus, we only need to classify which squares in the complementary board are deleted to make the derived complementary board. We start by deleting ℓ squares corresponding to the deletion of the first ℓ rows, namely $(1,1),(2,2),\ldots,(\ell,\ell)$.

Some of these squares may also be in columns $\alpha_1,\alpha_2,\ldots,\alpha_\ell$, but to avoid double-counting, we only consider those letters that are greater than ℓ. These are $|\{\ell+1,\ell+2,\ldots,n\}\cap\{\alpha_1,\alpha_2,\ldots,\alpha_\ell\}|$, as desired. □

4.4 Derangements with a given prefix

Now that we have a way of quickly computing $|B_\alpha^c|$, we can compute the number of ways to place a given number of rooks on the complementary board. We can use this to compute the rook polynomial for the derived complementary board $p_{B_\alpha^c}(x)$. We will see later that we can use the coefficients of this polynomial to compute the number of ways of placing $n-\ell$ rooks on the derived board B_α.

Lemma 20. *The rook polynomial for the complementary board B_α^c is*

$$p_{B_\alpha^c}(x)=\sum_{j=0}^{|B_\alpha^c|}\binom{|B_\alpha^c|}{j}x^j. \tag{12}$$

Proof. Recall that no two squares of B_α^c are in the same row or column. Thus the number of ways to place j rooks is equivalent to selecting any j squares from the collection of $|B_\alpha^c|$ squares.

Therefore the coefficient of x^j in the rook polynomial is $\binom{|B_\alpha^c|}{j}$. □

Now we introduce a lemma of Stanley [8] to compute the number of ways of placing $n-\ell$ rooks in the derived board $B_\alpha\subseteq[n-\ell]\times[n-\ell]$.

Lemma 21 ([8])**.** *Let $B\subseteq[n]\times[n]$ be a board with complementary board B^c, and denote the rook polynomial of B^c by $P_{B^c}(x)=\sum_{k=0}^n r_k^c x^k$.*

Then the number of ways, N_0, of placing n nonattacking rooks on B is given by the principle of inclusion/exclusion

$$N_0=\sum_{k=0}^{n}(-1)^k r_k^c (n-k)!. \tag{13}$$

This lemma allows us to compute the number of rook placements on the derived board B_α, which is the number of derangements in $\mathscr{D}$ that begin with the prefix α.

Corollary 22. *The number of derangements with prefix $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_\ell)$ is given by*

$$\#\,\mathrm{prefix}_{\mathscr{D}}(\alpha)=\sum_{j=0}^{|B_\alpha^c|}(-1)^j\binom{|B_\alpha^c|}{j}(n-\ell-j)!, \tag{14}$$

which is A047920$(n-\ell, |B^c_\alpha|)$ *in the On-Line Encyclopedia of Integer Sequences [4].*

Because we can compute $|B^c_\alpha|$ from α in linear time (see Lemma 19), if we use a computational model where factorials are given by an oracle and arithmetic can be computed in constant time, then $\#\,\mathrm{prefix}_{\mathscr{D}}$ can be computed in linear time with respect to ℓ, the length of the prefix.

Example 23. *For $n = 12$, we wish to count the number of derangements that start with the prefix $\alpha = (6, 1)$, as illustrated in Figure 2. Since the prefix has two letters, $\ell = 2$ and $n - \ell = 12 - 2 = 10$. The number of squares in B^c_α is*

$$|B^c_\alpha| = 12 - 2 - \underbrace{|\{3, 4, \dots, 12\} \cap \{6, 1\}|}_{1} = 9. \tag{15}$$

Thus there are A047920$(10, 9) = 1\,468\,457$ derangements in S_{12} that start with the prefix $\alpha = (6, 1)$.

Now that we have an efficient algorithm for computing $\#\,\mathrm{prefix}_{\mathscr{D}} \colon \mathcal{W} \to \mathbb{N}_{\geq 0}$, we can invoke the recursive formula in Theorem 7 to compute $\mathrm{unrank}_{\mathscr{D}} \colon \mathbb{N}_{\geq 0} \to \mathscr{D}$ and unrank derangements. The sequence of recursive steps is illustrated in Table 1.

5 Unranking ménage permutations

After claiming Richard's prize for unranking derangements, the conversation shifted to how this technique could be extended. A natural next step seemed to be to look at another family of restricted permutations, namely ménage permutations.

A ménage permutation comes from the *problème des ménages*, introduced by Édouard Lucas in 1891. There are a few choices of how to define these permutations, but we will use the following definition for simplicity.

Definition 24. *A **ménage permutation** is a permutation $\pi \in S_n$ such that for all $i \in [n]$, $\pi(i) \neq i$ and $\pi(i) + 1 \not\equiv i \bmod n$. The set of ménage permutations of length n is denoted by $\mathcal{M}_n$.*

5.1 The answer to a $100 question about ménage permutations

By February 2020, it appeared that the techniques for unranking derangements would not directly translate to the context of ménage permutations. In response, Richard Arratia upped the stakes by offering another prize for unranking ménage permutations. Specifically, he posed the following problem.

Problem 25. *For $n = 20$ there are A000179$(20) = 312\,400\,218\,671\,253\,762 > 3.1 \cdot 10^{17}$ ménage permutations. Determine the 10^{17}-th such permutation when listed in lexicographic order.*

α (prefix)	$\#\text{prefix}_{\mathscr{D}}(\alpha)$	index range	$\lvert B^c_\alpha \rvert$	$\text{unrank}_{\mathscr{D}}(1000)$
1	0	$(0, 0]$	–	$f^{\mathscr{D}}_{1000}(1, 0)$
2	2119	$(0, 2119]$	6	$f^{\mathscr{D}}_{1000}(2, 0)$
21	265	$(0, 265]$	6	$f^{\mathscr{D}}_{1000}(21, 0)$
22	0	$(265, 265]$	–	$f^{\mathscr{D}}_{1000}(22, 265)$
23	309	$(265, 574]$	5	$f^{\mathscr{D}}_{1000}(23, 265)$
24	309	$(574, 883]$	5	$f^{\mathscr{D}}_{1000}(24, 574)$
25	309	$(883, 1192]$	5	$f^{\mathscr{D}}_{1000}(25, 883)$
251	53	$(883, 936]$	4	$f^{\mathscr{D}}_{1000}(251, 883)$
253	0	$(936, 936]$	–	$f^{\mathscr{D}}_{1000}(253, 936)$
254	64	$(936, 1000]$	3	$f^{\mathscr{D}}_{1000}(254, 936)$
2541	11	$(936, 947]$	3	$f^{\mathscr{D}}_{1000}(2541, 936)$
2543	11	$(947, 958]$	3	$f^{\mathscr{D}}_{1000}(2543, 947)$
2546	14	$(958, 972]$	2	$f^{\mathscr{D}}_{1000}(2546, 958)$
2547	14	$(972, 986]$	2	$f^{\mathscr{D}}_{1000}(2547, 972)$
2548	14	$(986, 1000]$	2	$f^{\mathscr{D}}_{1000}(2548, 986)$
25481	3	$(986, 989]$	2	$f^{\mathscr{D}}_{1000}(25481, 986)$
25483	3	$(989, 992]$	2	$f^{\mathscr{D}}_{1000}(25483, 989)$
25486	4	$(992, 996]$	1	$f^{\mathscr{D}}_{1000}(25486, 992)$
25487	4	$(996, 1000]$	1	$f^{\mathscr{D}}_{1000}(25487, 996)$
254871	2	$(996, 998]$	0	$f^{\mathscr{D}}_{1000}(254871, 996)$
254873	2	$(998, 1000]$	0	$f^{\mathscr{D}}_{1000}(254873, 998)$
2548731	1	$(998, 999]$	0	$f^{\mathscr{D}}_{1000}(2548731, 998)$
2548736	1	$(999, 1000]$	0	$f^{\mathscr{D}}_{1000}(2548736, 999)$
25487361	1	$(999, 1000]$	0	$f^{\mathscr{D}}_{1000}(25487361, 999)$

Table 1: There are $A000166(8) = 14833$ derangements on 8 letters. The table shows the recursive steps to find that the derangement at index 1000 is 25487361.

Using the techniques described in this section, we developed a computer program that computed the answer in less than 30 milliseconds. (By comparison, unranking the 10^{157}-th ménage permutation of the more than 1.25×10^{157} ménage permutations in S_{100} with the same program takes about 7 seconds.)

Answer 26. *The desired permutation is*

$$7\ 16\ 19\ 12\ 2\ 8\ 15\ 1\ 18\ 14\ 3\ 9\ 20\ 10\ 5\ 17\ 13\ 4\ 11\ 6. \tag{16}$$

5.2 Overview for unranking ménage permutations

As in the section about unranking derangements, we will use the insight from Theorem 7 that if we can efficiently count the number of words with a given prefix, then we can efficiently unrank the words.

The technique exploits the following observations: after placing rooks on a board corresponding to our prefix, the remaining board has the property that its complement can be partitioned into sub-boards that do not share rows or columns. These sub-boards have a structure that we can understand, and we can leverage that understanding to compute the rook polynomials of these sub-boards and consequently of the complementary board itself. Once we have computed the rook polynomial of the complementary board, we can again use Lemma 21 to compute the number of full rook placements on the original board. This gives us the number of ménage permutations with a given prefix.

5.3 Disjoint board decomposition

Figure 3 suggestively shows a placement of rooks according to a prefix that results in a board whose complement can be partitioned into sub-boards whose squares don't share any rows or columns. We will see that this property indeed holds in general, and we can exploit this in order to count the ménage permutations with a given prefix.

The property of complements that can be partitioned into sub-boards whose squares don't share rows or columns is useful because it provides a way of factoring the rook polynomial of the bigger board into the rook polynomials of the sub-boards.

Definition 27. *Two sub-boards B and B' are called* ***disjoint*** *if no squares of B are in the same row or column as any square in B'.*

Kaplansky gives a way of computing the rook polynomial of a board in terms of its disjoint boards.

Theorem 28 ([1])**.** *If B can be partitioned into disjoint boards $b_1, b_2, \ldots, b_m$, then the rook polynomial of B is the product of the rook polynomials of each sub-board*

$$p_B(x) = \prod_{i=1}^{m} p_{b_i}(x). \tag{17}$$

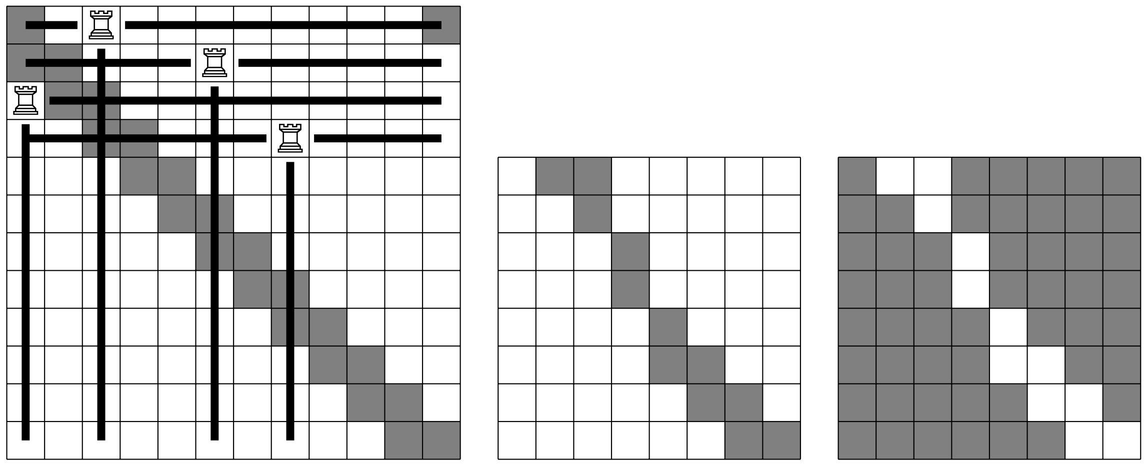


Figure 3: The prefix $\alpha = (3, 6, 1, 8)$, the derived board B_α, and the derived complementary board $B^c_\alpha = \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}^{\intercal}_7$. There are 8062 ways of placing eight nonattacking rooks on B_α.

We will use this disjoint board decomposition repeatedly, because the boards that result after placing a prefix can be partitioned into disjoint sub-boards whose structure is well understood. Now we will give a name to these blocks, which are illustrated in Figure 4.

Definition 29. *A board is called* ***staircase-shaped*** *if it matches one of the following four shapes:*

$$\begin{aligned} \mathscr{O}_{2n-1} &= \{(i,i) : i \in [n]\} &&\cup\ \{(i,i+1) : i \in [n-1]\} \\ \mathscr{O}^{\intercal}_{2n-1} &= \{(i,i) : i \in [n]\} &&\cup\ \{(i+1,i) : i \in [n-1]\} \\ \mathscr{E}_{2n-2} &= \{(i,i) : i \in [n-1]\} &&\cup\ \{(i+1,i) : i \in [n-1]\} \\ \mathscr{E}^{\intercal}_{2n-2} &= \{(i,i) : i \in [n-1]\} &&\cup\ \{(i,i+1) : i \in [n-1]\}. \end{aligned}$$

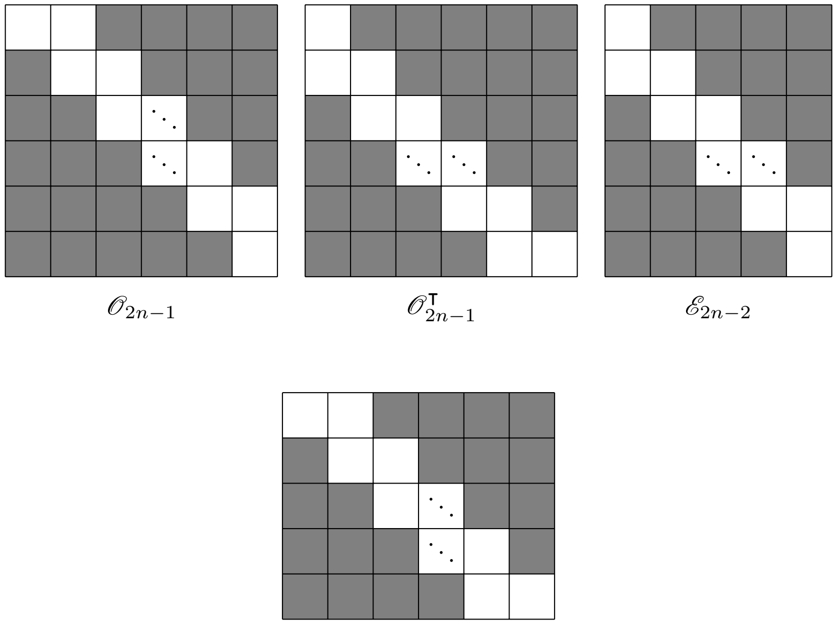
The subscripts represent the number of squares, and the names represent their parity.

We now show that our resulting boards can be partitioned into boards of these shapes.

Lemma 30. *For $\ell \geq 1$, and prefix $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ the derived complementary board B^c_α can be partitioned into disjoint staircase-shaped boards.*

Proof. The proof proceeds by induction on the length of the prefix.

To establish the base case, consider a prefix of length $\ell = 1$. Because of the ménage restriction, $\alpha_1 \in \{2, 3, \ldots, n-1\}$, and the derived complementary board $B^c_{(\alpha_1)}$ can be partitioned into two disjoint sub-boards with shapes $\mathscr{O}_{2\alpha_1-3}$ and $\mathscr{O}^{\intercal}_{2n-2\alpha_1-1}$. (This is illustrated for the case of $n = 7$ and α_1 in Figure 5.)



$\mathscr{O}_{2n-1}$ $\mathscr{O}^{\mathsf{T}}_{2n-1}$ $\mathscr{E}_{2n-2}$ $\mathscr{E}^{\mathsf{T}}_{2n-2}$

Figure 4: Examples of each of the four staircase-shaped boards. The first two boards are on grids of size $n \times n$, the third is on a grid of size $n \times (n-1)$ and the fourth is on a grid of size $(n-1) \times n$.

The inductive hypothesis is that the derived complementary board for a prefix of length $\ell - 1$ consists of sub-boards with shape $\mathscr{O}_{2m-1}$, $\mathscr{O}^{\mathsf{T}}_{2m-1}$, $\mathscr{E}_{2m-2}$, or $\mathscr{E}^{\mathsf{T}}_{2m-2}$. Placing a rook in row ℓ can remove a top row or a column or both in a given sub-board. Table 2 below shows the resulting sub-boards after placing a rook in ℓ-th row of B, which may be in the top row, the i-th column, or both.

Rook placement	$\mathscr{O}_{2m-1}$	$\mathscr{O}^{\mathsf{T}}_{2m-1}$	$\mathscr{E}_{2m-2}$	$\mathscr{E}_{2m-2}$
Row 1	$\mathscr{O}_{2m-3}$	$\mathscr{E}^{\mathsf{T}}_{2m-2}$	$\mathscr{O}_{2m-3}$	$\mathscr{E}^{\mathsf{T}}_{2m-4}$
Column i	$\mathscr{O}_{2i-3}$, $\mathscr{E}_{2m-2i}$	$\mathscr{E}_{2i-2}$, $\mathscr{O}^{\mathsf{T}}_{2m-2i-1}$	$\mathscr{E}_{2i-2}$, $\mathscr{E}_{2m-2i-2}$	$\mathscr{O}_{2i-3}$, $\mathscr{O}^{\mathsf{T}}_{2m-2i-1}$
Row 1, column i	$\mathscr{O}_{2i-5}$, $\mathscr{E}_{2m-2i}$	$\mathscr{O}_{2i-3}$, $\mathscr{O}^{\mathsf{T}}_{2m-2i-1}$	$\mathscr{O}_{2i-3}$, $\mathscr{E}_{2m-2i-2}$	$\mathscr{O}_{2i-5}$, $\mathscr{O}^{\mathsf{T}}_{2m-2i-1}$

Table 2: The results of placing a rook in the first row, i-th column, or both for all staircase-shaped boards.

Therefore placing any number of rooks in the first ℓ rows results in a board whose complementary derived board is composed of disjoint staircase-shaped sub-boards. □

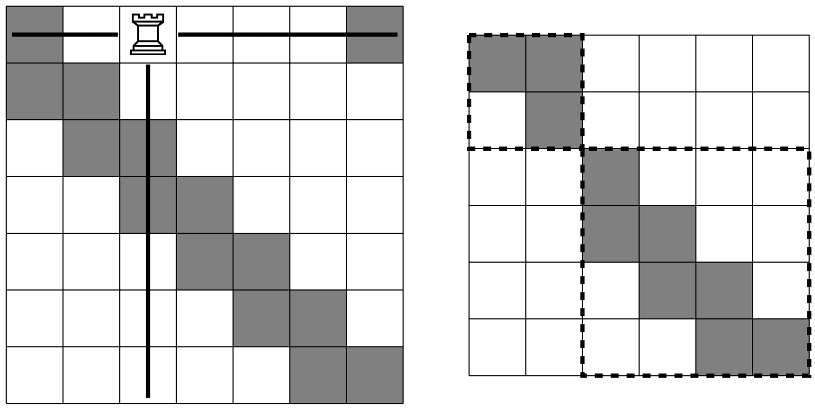


Figure 5: The first chessboard shows a placement of a rook at position 3, the second shows how the derived complementary board can be partitioned into two disjoint boards with 3 and 7 squares respectively.

5.4 Rook polynomials of blocks

Recall that the goal of partitioning B into disjoint sub-boards $b_1, b_2, \ldots, b_m$ is so that we can factor $p_B(x)$ in terms of $p_{b_i}(x)$. Of course, this is only useful if we can describe $p_{b_i}(x)$, which is the goal of this subsection. Conveniently, the rook polynomial of each b_i will turn out to depend only on the number of squares, $|b_i|$, which can be computed recursively because of its staircase shape.

We will begin by defining a family of polynomials that, suggestively, will turn out to be the rook polynomials that we are looking for. (The coefficients of these polynomials are described by OEIS sequence A011973 [4].)

Definition 31. *For $j \geq 0$, the j-th **Fibonacci polynomial** $F_j(x)$ is defined recursively as:*

$$F_0(x) = 1 \tag{18}$$
$$F_1(x) = 1 + x \tag{19}$$
$$F_n(x) = xF_{n-2}(x) + F_{n-1}(x). \tag{20}$$

The rook polynomials of the staircase-shaped boards agree with these Fibonacci polynomials.

Lemma 32. *If B is a staircase-shaped board with k squares, then B has rook polynomial $p_B(x) = F_k(x)$, equal to the k-th Fibonacci polynomial.*

Proof. We will recall the recursive construction of rook polynomials from Lemma 14, and proceed by induction on the number of squares, always choosing to include or exclude the upper-left square.

Since the reflections of board have the same rook polynomial as the unreflected board, without loss of generality, we will compute the rook polynomials for $\mathscr{O}_{2m-1}$ and $\mathscr{E}_{2m-2}$, respectively.

To establish a base case, consider the rook polynomials when $n = 1$, so the even board has $|\mathscr{E}_0| = 0$ squares and the odd board has $|\mathscr{O}_1| = 1$ square. We can see the corresponding rook polynomials directly. There is 1 way to place 0 rooks on $\mathscr{E}_0$ and no ways to place more rooks; similarly there is 1 way to place 0 rooks on $\mathscr{O}_1$, 1 way to place 1 rook on $\mathscr{O}_1$, and no way to place more than one rook. Thus

$$p_{\mathscr{E}_0}(x) = 1 \qquad = F_0(x), \text{ and} \tag{21}$$
$$p_{\mathscr{O}_1}(x) = 1 + x = F_1(x). \tag{22}$$

With the base case established, our inductive hypothesis is that $p_B(x) = F_h(x)$ whenever B is a staircase-shaped board with $h < k$ squares.

Assume that we have k squares where k is even, so our board looks like $\mathscr{E}_k$. We can either place a rook or not in the upper-left square. If we include the square, then $(\mathscr{E}_k)_i \cong \mathscr{E}_{k-2}$, if we exclude the square, then $(\mathscr{E}_k)_e \cong \mathscr{O}_{k-1}$. Thus by Lemma 14, the rook polynomial of $\mathscr{E}_k$ is

$$p_{\mathscr{E}_k}(x) = x p_{\mathscr{E}_{k-2}}(x) + p_{\mathscr{O}_{k-1}}(x) \tag{23}$$
$$= xF_{k-2}(x) + F_{k-1}(x) \tag{24}$$
$$= F_k(x). \tag{25}$$

The case where k is odd proceeds in almost the same way. Here our board looks like $\mathscr{O}_k$. We can either place a rook or not in the upper-left square. If we include the square, then $(\mathscr{O}_k)_i \cong \mathscr{O}_{k-2}$, if we exclude the square, then $(\mathscr{O}_k)_e \cong \mathscr{E}_{k-1}$. Again by Lemma 14, the rook polynomial of $\mathscr{O}_k$ is

$$p_{\mathscr{O}_k}(x) = x p_{\mathscr{O}_{k-2}}(x) + p_{\mathscr{E}_{k-1}}(x) \tag{26}$$
$$= xF_{k-2}(x) + F_{k-1}(x) \tag{27}$$
$$= F_k(x). \tag{28}$$

□

Therefore, we now have the ingredients to describe the rook polynomial of a derived complementary board.

Corollary 33. *Suppose that B_α^c can be partitioned into m disjoint staircase-shaped sub-boards of sizes $b_1, b_2, \ldots, b_m$. Then the rook polynomial of B_α^c is*

$$p_{B_\alpha^c}(x) = \prod_{i=1}^{m} F_{b_i}, \tag{29}$$

where F_j is the j-th Fibonacci polynomial.

Proof. This follows directly from Theorem 28 together with Lemma 32. □

5.5 Sub-boards from prefix

In this part, we discuss how to algorithmically compute the size of the sub-boards of the partition of the derived complementary board B^c_α for a given prefix α.

Lemma 34. *Given a nonempty prefix $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_\ell)$ and $i \notin \alpha$, the number of squares of B^c in column i that do not have a first coordinate in $[\ell]$ is given by the rule:*

$$c_i = \begin{cases} 0 & i < \ell \\ 1 & i = \ell \text{ or } i = n \\ 2 & \ell < i < n \end{cases} \tag{30}$$

Proof. It is helpful to recall that the complementary board B^c consists of squares on the diagonal, squares on the subdiagonal, and the square $(1, n)$:

$$B^c = \{(i,i) : i \in [n]\} \cup \{(i+1, i) : i \in [n-1]\} \cup \{(1,n)\}. \tag{31}$$

Now if $i < \ell$, then (i,i) and $(i+1,i)$ both have a first coordinate less than or equal to ℓ.

If $i = \ell$, then (i,i) has a first coordinate in $[\ell]$, but $(i+1,i) = (\ell+1,\ell)$ does not have its first coordinate in $[\ell]$.

If $i = n$, there are two squares of B^c in column i: (n,n) and $(1,n)$. Only $(1,n)$ has its first coordinate in $[\ell]$.

If $\ell < i < n$, then neither the square (i,i) nor $(i+1,i)$ has its first coordinate in $[\ell]$. □

Now we will go through each contiguous section of columns, and count the number of squares in each to build up the size of each of the blocks.

Lemma 35. *Partition $[n] \setminus \alpha$ into contiguous parts, $\mathcal{P}$. Each part $P_i \in \mathcal{P}$ of the partition corresponds to a staircase-shaped sub-board of size $\sum_{p \in P_i} c_p$.*

Therefore the size of the disjoint sub-boards in the derived complementary board B^c_α is given by the multiset

$$\mathcal{P}_\alpha = \left\{ \sum_{p \in P_i} c_p : P_i \in \mathcal{P} \right\}. \tag{32}$$

Proof. Once the first row of a complementary ménage board has been deleted, the resulting board has the property that any two nonadjacent columns do not have any squares in the same row, because column i has squares in (i,i) and $(i+1,i)$.

Within each contiguous interval between the letters of α, the columns form a staircase-shaped sub-board because each column with a square in position $(i+1,i)$ has a square to its right, in position $(i+1,i+1)$ whenever $i+1 \notin \alpha$. □

Example 36. *As illustrated in Figure 3, if* $n = 12$ *and* $\alpha = (3, 6, 1, 8)$*, then the contiguous partition of*

$$[12] \setminus \{3, 6, 1, 8\} = \{\underbrace{2}_{P_1}, \underbrace{4, 5}_{P_2}, \underbrace{7}_{P_3}, \underbrace{9, 10, 11, 12}_{P_4}\} \tag{33}$$

is $\{P_1, P_2, P_3, P_4\}$*. The corresponding staircase-shaped sub-boards have sizes*

$$\begin{aligned} k_1 &= c_2 &&= 0 &&= 0 \\ k_2 &= c_4 + c_5 &&= 1 + 2 &&= 3 \\ k_3 &= c_7 &&= 2 &&= 2 \\ k_4 &= c_9 + c_{10} + c_{11} + c_{12} &&= 2 + 2 + 2 + 1 &&= 7, \end{aligned}$$

which matches what we observe in the illustration:

$$B_\alpha^c = \mathscr{E}_0 \sqcup \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}_7^\intercal, \tag{34}$$

5.6 Complementary polynomials

We have now established a method taking a prefix α and partitioning B_α^c into disjoint staircase-shaped sub-boards, which allow us to determine the rook polynomial of B_α^c. Using Lemma 21, this allows us to finally compute the number of ways of placing $n - \ell$ rooks on B_α, thus determining the number of derangements that begin with α.

Theorem 37. *The number of ménage permutations that begin with a valid, nonempty prefix* α *is*

$$\#\,\text{prefix}_{\mathcal{M}}(\alpha) = \sum_{k=0}^{n} (-1)^k r_k^c (n-k)! \tag{35}$$

where $\sum_{k=0}^{n} r_k^c x^k = \prod_{p \in \mathcal{P}_\alpha} F_p$*,* F_k *is the* k*-th Fibonacci polynomial, and* $\mathcal{P}_\alpha$ *is the multiset corresponding to the size of the staircase-shaped sub-boards in the disjoint partition of* B_α^c*.*

Proof. This follows directly from Corollary 33 together with Lemma 35 ☐

Now that we have computed the number of ménage permutations, Theorem 7 provides an efficient unranking algorithm for $\mathcal{M}$.

We will illustrate this with a specific example computing the number of ménage permutations with a given prefix.

Example 38. *We will continue with the running example illustrated in Figure 3 and expounded on in Example 36.*

We've already seen that the for $n = 12$, the prefix $\alpha = (3, 6, 1, 8)$ partitions the derived complementary board into three nonempty sub-boards:

$$B_\alpha^c = \mathscr{E}_0 \sqcup \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}_7^\intercal. \tag{36}$$

Lemma 32 tells us that the rook polynomial of B_α^c is

$$\begin{aligned} p_{B_\alpha^c}(x) &= F_3(x)F_2(x)F_7(x) & (37)\\ &= (1 + 3x + x^2)(1 + 2x)(1 + 7x + 15x^2 + 10x^3 + x^4) & (38)\\ &= 1 + 12x + 57x^2 + 136x^3 + 170x^4 + 105x^5 + 27x^6 + 2x^7 & (39)\\ &= \sum_{k=0}^{7} r_k^c x^k. & (40) \end{aligned}$$

By Lemma 21, the number of ways to place eight rooks on B_α is is

$$\begin{aligned} N_0 &= \sum_{k=0}^{7} (-1)^k r_k^c (8-k)! & (41)\\ &= 1(8!) - 12(7!) + 57(6!) - 136(5!) + 170(4!) - 105(3!) + 27(2!) - 2(1!) & (42)\\ &= 8062. & (43) \end{aligned}$$

Therefore there are 8062 ménage permutations in S_{12} that start with the prefix $(3, 6, 1, 8)$.

We can now repeatedly use the above counting technique in conjunction with Theorem 7 to unrank derangements.

Example 39. *There are $A000179(8) = 4738$ ménage permutations on 8 letters. Table 3 shows the steps of the algorithm that determines that the 1000th ménage permutation in lexicographic order is*

$$w_{1000} = 3\ 5\ 4\ 8\ 2\ 7\ 1\ 6. \tag{44}$$

6 Generalizations and open questions

In this final section we explore several possible future directions for applying these ideas in new contexts. We can potentially apply these unranking techniques to position-restricted permutations, permutations that satisfy certain inequalities with respect to a permutation statistic, or words that avoid or match certain patterns.

6.1 Other restricted permutations

In a 2014 paper about finding linear recurrences for derangements, ménage permutations and other restricted permutations, Doron Zeilberger introduces a more general family of restricted permutations.

α	# prefix(α)	index range	block sizes	unrank$_{\mathcal{M}}(i)$
1	0	$(0,0]$	–	$f_{1000}^{\mathcal{M}}(1,0)$
2	787	$(0,787]$	$(1,11)$	$f_{1000}^{\mathcal{M}}(2,0)$
3	791	$(787,1578]$	$(3,9)$	$f_{1000}^{\mathcal{M}}(3,787)$
31	0	$(787,787]$	–	$f_{1000}^{\mathcal{M}}(31,787)$
32	0	$(787,787]$	–	$f_{1000}^{\mathcal{M}}(32,787)$
33	0	$(787,787]$	–	$f_{1000}^{\mathcal{M}}(33,787)$
34	159	$(787,946]$	$(1,7)$	$f_{1000}^{\mathcal{M}}(34,787)$
35	166	$(946,1112]$	$(1,2,5)$	$f_{1000}^{\mathcal{M}}(35,946)$
351	24	$(946,970]$	$(0,2,5)$	$f_{1000}^{\mathcal{M}}(351,946)$
...	0	$(970,970]$	–	
354	34	$(970,1004]$	$(0,5)$	$f_{1000}^{\mathcal{M}}(354,970)$
3541	5	$(970,975]$	$(0,5)$	$f_{1000}^{\mathcal{M}}(3541,970)$
3542	5	$(975,980]$	$(0,5)$	$f_{1000}^{\mathcal{M}}(3542,975)$
...	0	$(980,980]$	–	
3546	8	$(980,988]$	$(0,3)$	$f_{1000}^{\mathcal{M}}(3546,980)$
3547	10	$(988,998]$	$(0,2,1)$	$f_{1000}^{\mathcal{M}}(3547,988)$
3548	6	$(998,1004]$	$(0,4)$	$f_{1000}^{\mathcal{M}}(3548,998)$
35481	1	$(998,999]$	$(0,4)$	$f_{1000}^{\mathcal{M}}(35481,998)$
35482	1	$(999,1000]$	$(0,4)$	$f_{1000}^{\mathcal{M}}(35482,999)$
354821	0	$(999,999]$	(3)	$f_{1000}^{\mathcal{M}}(354821,999)$
...	0	$(999,999]$	–	
354827	1	$(999,1000]$	$(0,1)$	$f_{1000}^{\mathcal{M}}(354827,999)$
3548271	1	$(999,1000]$	(0)	$f_{1000}^{\mathcal{M}}(3548271,999)$
35482716	1	$(999,1000]$	$()$	$f_{1000}^{\mathcal{M}}(35482716,999)$

Table 3: The recursive computation of the 1000th ménage permutation.

Definition 40 ([10])**.** *Let $S \subset \mathbb{Z}$ be a finite collection of integers. An **S-avoiding permutation** is a permutation $\pi \in S_n$ such that*

$$\pi(i) - i - s \not\equiv 0 \bmod n \quad \text{for all} \quad i \in [n] \quad \text{and} \quad s \in S. \tag{45}$$

Example 41. *In terms of S-avoiding permutations,*

- *ordinary permutations are $\emptyset$-avoiding permutations,*
- *derangements are $\{0\}$-avoiding permutations, and*
- *ménage permutations are $\{-1, 0\}$-avoiding permutations.*

The results in the previous sections straightforwardly adapt to the cases of unranking $\{i\}$-avoiding and $\{i, i+1\}$ avoiding permutations.

Open Question 42. *For arbitrary finite subsets $S \subset \mathbb{Z}$, do there exist efficient unranking algorithms on S-avoiding permutations?*

The techniques used to unrank derangements and ménage permutations do not appear to generalize even to superficially similar domains. So, in the spirit of Richard Arratia's bounties, it is only fair to offer one of my own.

Problem 43. *Do there exist efficient unranking algorithms on $\{-1, 1\}$-avoiding permutations?*

The main obstruction to using the techniques from Section 5 to resolve this question is that placing a rook and deleting a column does not necessarily cause the left and right sides of that column to be disjoint. As such, unranking $\{-1, 1\}$-avoiding permutations appears to require a genuinely novel insight.

6.2 Permutation statistics

Another area for exploration is unranking permutations with a given permutation statistic.

Open Question 44. *Let $\operatorname{inv}\colon S_n \to \mathbb{N}_{\geq 0}$ be the map that counts inversions of a permutation. Since inv is a Mahonian statistic, the generating function for the number of permutations $\pi \in S_n$ such that $\operatorname{inv}(\pi) = k$ is given by the q analog of $n!$, $n!_q$.*

Does there exist an efficient unranking function on the set

$$\mathscr{I}_n^k = \{\pi \in S_n \,|\, \operatorname{inv}(\pi) = k\}, \tag{46}$$

and if so, how does one construct it?

We can, of course, substitute inv with any other permutation statistic of interest.

6.3 Pattern avoidance

In the field of combinatorics on words, there exists a notion of patterns and instances of a pattern. At this level of informality, this is probably best illustrated with an example (with undefined words in bold).

Example 45. *The word* 1100010110 *is an* ***instance*** *of the* ***pattern*** ABA *with* $A = 110$ *and* $B = 0010$*. The word* 32123213212 *is said to* ***match*** *the pattern* CC *with* $C = 321$ *because it contains a substring of the form* 321321*.*

Open Question 46. *Given a pattern* P*, is it possible to unrank words of length* n *over an alphabet* $\mathscr{A}$ *that are not instances of the pattern* P*? That match the pattern* P*? That don't match the pattern* P*?*

6.4 Prefixes of Lyndon words

There are other collections of finite words that might be amenable to some of the above techniques. In particular, Kociumaka, Radoszewski, and Rytter [2] give polynomial time algorithms for unranking Lyndon words. We have some conjectures about prefixes of Lyndon words and open questions about other restricted words.

Definition 47. *A* ***Lyndon word*** *is a string over an alphabet of letters that is the unique minimum with respect to all of its rotations.*

Example 48. 00101 *is a Lyndon word because*

$$00101 = \min\{00101, 01010, 10100, 01001, 10010\} \tag{47}$$

is the unique minimum of all of its rotations.

011011 *is not a Lyndon word because while*

$$011011 = \min\{011011, 110110, 101101, 011011, 110110, 101101\}, \tag{48}$$

it is not the ***unique*** *minimum. (That is, rotating it three positions returns it to itself.)*

Definition 49 ([7])**.** *Suppose that* $\{a_i\}_{i=1}^{\infty}$ *and* $\{b_i\}_{i=1}^{\infty}$ *are integer sequences related by*

$$1 + \sum_{n=1}^{\infty} b_n x^n = \prod_{i=1}^{\infty} \frac{1}{(1-x^i)^{a_i}}. \tag{49}$$

Then $\{b_i\}_{i=1}^{\infty}$ *is said to be the* ***Euler transform*** *of* $\{a_i\}_{i=1}^{\infty}$*, denoted* $\mathcal{E}(\{a_i\}_{i=1}^{\infty}) = \{b_i\}_{i=1}^{\infty}$*.*

Definition 50. *Let* $\mathscr{L}_\alpha = \{\ell_n^\alpha\}_{n=1}^{\infty}$ *where* ℓ_n^α *is the number of Lyndon words with prefix* α *and length* n *over the alphabet* $\{0, 1\}$*.*

Conjecture 51. *The Euler transform of the number of Lyndon words with prefix* α *and length* n *over the alphabet* $\{0, 1\}$*,* $\mathcal{E}(\mathscr{L}_\alpha) = \{t_n^\alpha\}_{n=1}^{\infty}$*, follows a linear recurrence for all* $n \geq N_\alpha$*.*

This conjecture and the following conjectures are based on the data in Table **??**.

We start with two specific conjectures about two families of prefixes.

Conjecture 52. *For $k \geq 1$, let $\mathscr{L}_\alpha$ be the sequence of the number of Lyndon words of length n with prefix $\alpha = (0, 0, \ldots, 0)$ of length k over the alphabet $\{0, 1\}$. Then the Euler transform $\mathcal{E}(\mathscr{L}_\alpha) = \{t_n^\alpha\}_{n=1}^\infty$ follows the linear recurrence $t_{n+1}^\alpha = 2t_n^\alpha$ for all $n \geq k + 2$.*

Conjecture 53. *For $k \geq 2$ let $\mathscr{L}_\alpha$ be the sequence of the number of Lyndon words of length n with prefix $\alpha = (1, 0, 0, \ldots, 0)$ of length k over the alphabet $\{0, 1\}$. Then the Euler transform $\mathcal{E}(\mathscr{L}_\alpha) = \{t_n^\alpha\}_{n=1}^\infty$ follows the linear recurrence $t_{n+k}^\alpha = \sum_{i=0}^{k-1} t_{n+i}^\alpha$ for all $n \geq 1$.*

And more broadly, we have a conjecture in the case that the prefix is not the zero sequence.

Conjecture 54. *Let $\mathscr{L}_\alpha$ be the sequence of the number of Lyndon words of length n with prefix α over the alphabet $\{0, 1\}$ such that α contains at least one 1. Then the Euler transform of the sequence, $\mathcal{E}(\mathscr{L}_\alpha) = \{t_n^\alpha\}_{n=1}^\infty$, follows the linear recurrence where all terms have coefficients of 0 or 1.*

Open Question 55. *If, as the evidence suggests, $\mathcal{E}(\mathscr{L}_\alpha)$ follows a linear recurrence, what is the length of the recurrence and what are the coefficients of the recurrence as a function of α?*

References

[1] Irving Kaplansky and John Riordan. "The problem of the rooks and its applications". In: *Duke Mathematical Journal* 13.2 (1946), pp. 259–268. DOI: 10.1215/S0012-7094-46-01324-5. URL: https://doi.org/10.1215/S0012-7094-46-01324-5.

[2] Tomasz Kociumaka, Jakub Radoszewski, and Wojciech Rytter. "Computing k-th Lyndon Word and Decoding Lexicographically Minimal de Bruijn Sequence". In: *Combinatorial Pattern Matching*. Springer International Publishing, 2014, pp. 202–211. ISBN: 978-3-319-07566-2.

[3] Kenji Mikawa and Ken Tanaka. "Lexicographic ranking and unranking of derangements in cycle notation". In: *Discret. Appl. Math.* 166 (2014), pp. 164–169.

[4] OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*. 2021. URL: https://oeis.org/.

[5] John Riordan. *An Introduction to Combinatorial Analysis*. USA: Princeton University Press, 1980. ISBN: 9781400854332.

[6] V. S. Shevelev. “Some problems of the theory of enumerating the permutations with restricted positions”. In: *Journal of Soviet Mathematics* 61.4 (Sept. 1992), pp. 2272–2317. ISSN: 1573-8795. DOI: `10.1007/BF01104103`. URL: `https://doi.org/10.1007/BF01104103`.

[7] Neil James Alexander Sloane and Simon Plouffe. *Encyclopedia of Integer Sequences*. San Diego: Academic Press, 1995.

[8] Richard P. Stanley. *Enumerative Combinatorics: Volume 1*. 2nd. USA: Cambridge University Press, 2011. ISBN: 1107602629.

[9] L.G. Valiant. “The complexity of computing the permanent”. In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201. ISSN: 0304-3975. DOI: `https://doi.org/10.1016/0304-3975(79)90044-6`. URL: `https://www.sciencedirect.com/science/article/pii/0304397579900446`.

[10] Doron Zeilberger. “Automatic Enumeration of Generalized Ménage Numbers”. In: *Séminaire Lotharingien de Combinatoire* 71.B71a (2014).