Table of Contents

Chapter 1

Unranking restricted permutations

In this chapter we discuss efficient ways of 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.

1.1 Overview and history

In January 2020, Richard Arratia sent out an email announcing a talk he was going to give about "unranking derangements". He asked, if you list all derangements on some number of letters, listed as words in lexicographic order, we know how to enumerate them in both senses of the word: we can count them, and we list them out. However, he asked, what if we want to sample from a particular index somewhere in the middle of the list? Can you do this efficiently?

Having shown Richard a proof of concept for unranking derangements, he widened his next to a problem that I suspected was too hard. In January 2021, he announced a $100 prize for solving the analogous problem with ménage permutations.

Richard was, of course, interested in the more general question: given some family of combinatorial objects $\mathscr{C}_n$ that can be counted efficiently (for example, derangements or Dyck paths or graphs on *n* vertices) and some total ordering on them (for example, lexicographic order), when is it possible to **unrank** the collection in some computationally efficient way?

Definition 1.1.1. *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 $1 \leq i < |\mathscr{C}|$.*

*The **ranking map** is the map $\operatorname{rank}_{\mathscr{C}} \colon \mathscr{C} \to \mathbb{N}_{>0}$ which sends $c_i \mapsto i$.*

*The **unranking map** is the inverse map $\operatorname{unrank}_{\mathscr{C}} \colon \mathbb{N}_{>0} \to \mathscr{C}$ which sends $i \mapsto c_i$.*

In abstract terms, these maps are not particularly interesting, but in practical terms it can be quite difficult to efficiently compute a given ranking or unranking from a given totally ordered set. After all, when these sets grow in exponential time or worse, explicitly constructing the sequence and doing a search is not computationally feasible.

In Appendix **??**, we provide examples of totally ordered combinatorial objects $\mathscr{C}$ that have efficient ranking and unranking maps.

In this chapter we're going to explore this idea in the context of combinatorial words in lexicographic order. In particular, we will focus on two families of restricted permutations: derangements and ménage permutations. 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.

1.2 Prefix counting and word ranking

Proposition 1.2.1. *If we have an efficient way to compute the unranking map, an efficient way to compare two elements in the total order, and an efficient way of computing the number of objects at hand, $|\mathscr{C}|$, then we can efficiently compute the ranking map.*

Proof. The assumptions of the proposition are enough to perform a binary search over $\mathscr{C}$, which will contribute a factor of at worst $\log(|\mathscr{C}|)$ to the running time. $\square$

1.2.1 Counting words with a given prefix

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

Definition 1.2.2. *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 $\mathscr{W}_{\mathscr{A}}$, or just $\mathscr{W}$ when the alphabet is implicit from context.

Definition 1.2.3. *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 1.2.4. *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, $w_i < w'_i$.*

Lemma 1.2.5. *Let $\mathscr{W}$ be the set of words of any length on the alphabet $[n]$, and let $\mathscr{C} \subsetneq \mathscr{W}$ be a finite subset of words on this alphabet, with a total order equal to its lexicographic order.*

Then let $\#\mathrm{prefix}_{\mathscr{C}} \colon \mathscr{W} \to \mathscr{C}$ 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

$$\mathrm{unrank}_{\mathscr{C}}(i) = f_i^{\mathscr{C}}((1), 0) \tag{1.1}$$

where

$$f_i^{\mathscr{C}}(\alpha, j) = \begin{cases} \alpha & \alpha \in \mathscr{C} \text{ and } i = j+1 & \text{(1.2a)} \\ f_i^{\mathscr{C}}(\alpha', j) & \alpha \notin \mathscr{C} \text{ and } i \in (j, j + \#\mathrm{prefix}_{\mathscr{C}}(\alpha)] & \text{(1.2b)} \\ f_i^{\mathscr{C}}(\alpha'', j + \#\mathrm{prefix}_{\mathscr{C}}(\alpha)) & \text{otherwise,} & \text{(1.2c)} \end{cases}$$

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

Proof. We will start by inductively showing that j the number of words in $\mathscr{C}$ that occur strictly before α. The base case is clear: 0 words occur strictly before $\alpha = (1)$, because 1 is the lexicographically earliest nonempty word.

In Equation 1.2b, α' is α with a 1 appended. Since 1 is a minimal element and $\alpha \notin \mathscr{C}$, there are the same number of words preceding α as there are preceding α'.

In Equation 1.2c, the words that strictly precede α'' are the words that precede α, plus the words that begin with α, namely $j + \#\text{prefix}_{\mathscr{C}}(\alpha)$ of them.

Now, we go case by case in showing that the recursive behavior is correct.

To start, if $\alpha \notin \mathscr{C}$, but $i \in (j, j + \#\text{prefix}_{\mathscr{C}}(\alpha)]$, the range of indices of words that have prefix α, then the known letters of α are the prefix, and so we must add on more letters to determine the rest of the i-th word.

Otherwise we know that i is not in the range of indices with that starting prefix so we must increment the last letter in the prefix.

Lastly, if $\alpha \in \mathscr{C}$ and $i = j + 1$, then there are j words preceding α, so α is the i-th word, as desired. $\square$

This shows that if we can construct a function $\#\text{prefix}_{\mathscr{C}}$ that efficiently counts the number of elements of $\mathscr{C}$ with a given prefix, then we can efficiently rank and unrank the elements of $\mathscr{C}$ in lexicographic order.

1.2.2 Ranking words

In Proposition 1.2.1, we showed that given an efficient algorithm to compute $\text{unrank}_{\mathscr{C}}$, we can derive an efficient algorithm to compute $\text{rank}_{\mathscr{C}}$.

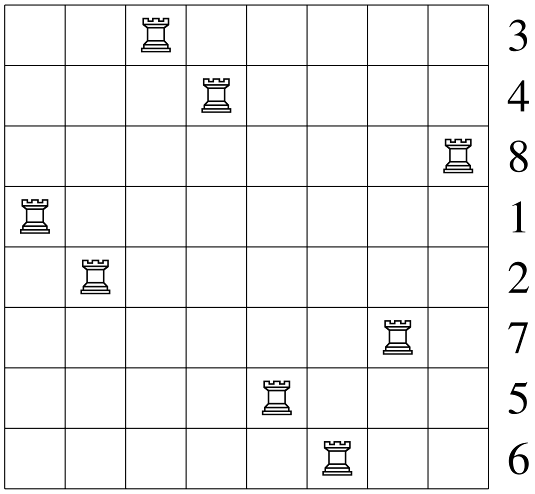

Figure 1.1: An illustration of the rook placement corresponding to the permutation $34812756 \in S_8$. A rook is placed in square $(i, \pi(i))$ for each i.

It is worth noting, however, that we can produce a more efficient algorithm, by ranking the given word recursively. In particular, we can say $\text{rank}_{\mathscr{C}}(w) = g_w(1,1,0)$ where

$$g_w(i,\ell,c) = \begin{cases} c+1 & \ell = w_i \text{ and } i = |w| \\ g_w(i+1,1,c) & \ell = w_i \text{ and } i < |w| \\ g_w(i,\ell+1,c+\#\text{prefix}_{\mathscr{C}}(w')) & \ell < w_i, \end{cases} \tag{1.3}$$

where $w' = (w_1, w_2, \ldots, w_{i-1}, \ell)$.

1.3 Basic notions of rook theory

Now that we have showed that we can unrank words if we can compute the number of words with a given prefix, we must 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.

1.3.1 Definitions in rook theory

We begin by introducing some preliminary ideas from rook theory.

Definition 1.3.1. *A **board** B is a subset of $[n] \times [n]$ which represents the squares of a $n \times n$ 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 valid squares in such a way that no two rooks are in the same row or column.

Definition 1.3.2. *The **rook polynomial** associated with a board B,*

$$p_B(x) = r_0 + r_1 x + r_2 x^2 + \cdots + r_n x^n, \tag{1.4}$$

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 do 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 computationally 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. (In fact, Shevelev [2] claims that "the theory of enumerating the permutations with restricted positions stimulated the development of the theory of the permanent.")

Lemma 1.3.3. *Let $M_B = \{a_{ij}\}$ be an $n \times n$ matrix where*

$$a_{ij} = \begin{cases} 1 & (i,j) \in B \\ 0 & (i,j) \notin B \end{cases}. \tag{1.5}$$

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

$$\operatorname{perm}(M_B) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}. \tag{1.6}$$

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

Theorem 1.3.4 (Valiant's Theorem [3])**.** *The counting problem of of computing the permanent of a (0,1)-matrix is #P-complete.*

Corollary 1.3.5. *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.

1.3.2 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 1.3.6 ([4])**.** *Given a board, B, then for any square $(x,y) \in B$, we can define the resulting boards if we include or exclude the square respectively*

$$B_i = \{(x',y') \in B : x \neq x' \text{ and } y \neq y'\} \tag{1.7}$$

$$B_e = B \setminus (x,y). \tag{1.8}$$

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

$$p_B(x) = x p_{B_i}(x) + p_{B_e}(x). \tag{1.9}$$

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. However, when the number of squares in B^c is small in some sense, it can be easier to compute the rook polynomial p_{B^c} and use the principle of inclusion/exclusion on it's 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. 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.

1.4 Unranking derangements

1.4.1 The answer to a \$100 question about derangements

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

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

\$100 Answer. *The computer program in Appendix* **??** *computed the answer in less than ten milliseconds. When written as words in lexicographic order, the derangement in* S_{20} *with rank* 5×10^{17} *is*

$$12\ 14\ 2\ 9\ 13\ 20\ 6\ 3\ 1\ 17\ 5\ 11\ 19\ 15\ 10\ 18\ 8\ 7\ 4\ 16. \tag{1.10}$$

1.4.2 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 [5] give an algorithm to rank/unrank derangements with respect to *lexicographic ordering in cycle notation.*

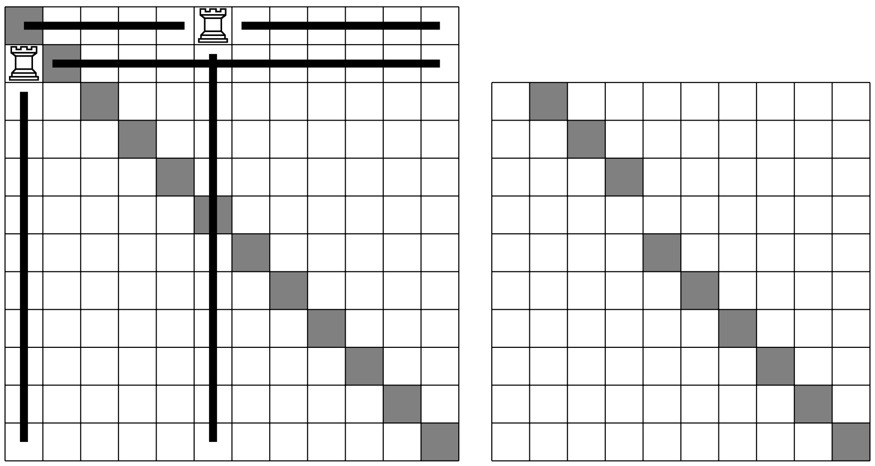

Figure 1.2: An example of a prefix $\alpha = (6,1)$, and the board that results from deleting the first $\ell = 2$ rows and columns 6 and 1. The derived complementary board of B from α is $B^c_\alpha = \{(1,2),(2,3),(3,4),(5,5),\ldots,(10,10)\}$.

In this section we will develop an algorithm for ranking and unranking 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 1.4.1. *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*

$$\mathscr{D}_n = \{\pi \in S_n : \pi(i) \neq i \; \forall i \in [n]\}. \tag{1.11}$$

1.4.3 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 k rooks according to these positions, as illustrated in Figure 1.2.

Definition 1.4.2. *If B is an $n \times n$ board, and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ is a valid prefix of length ℓ, then **derived board** of B from α, denoted B_α, is constructed by removing rows $1,2,\ldots,\ell$ and columns $\alpha_1, \alpha_2, \ldots, \alpha_e ll$ 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 1.4.3. *Given a valid ℓ letter prefix $(\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ of a word on n letters, the number of squares in the derived complementary board is*

$$|B^c_\alpha| = n - \ell - |\{\ell+1, \ell+2, \ldots, n\} \cap \{\alpha_1, \alpha_2, \ldots, \alpha_\ell\}|, \tag{1.12}$$

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. □

1.4.4 Derangements with a given prefix

Now that we have a way of quickly computing $|B^c_\alpha|$, 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 $p_{B^c_\alpha}(x)$. 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 1.4.4. *The rook polynomial for the complementary board B_α^c is*

$$p_{B_\alpha^c}(x) = \sum_{j=0}^{|B_\alpha^c|} \binom{|B_\alpha^c|}{j} x^j. \tag{1.13}$$

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 [6] to compute the number of ways of placing $n-\ell$ rooks in a complementary board $B_\alpha \subseteq [n-\ell] \times [n-\ell]$.

Lemma 1.4.5 ([6]). *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

$$N_0 = \sum_{k=0}^{n} (-1)^k r_k^c (n-k)!. \tag{1.14}$$

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

Corollary 1.4.6. *The number of derangements with prefix $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ is given by*

$$\#\text{prefix}_{\mathscr{D}}(\alpha) = \sum_{j=0}^{|B_\alpha^c|} (-1)^j \binom{|B_\alpha^c|}{j} (n-\ell-j)!, \tag{1.15}$$

which is A047920$(n-\ell, |B_\alpha^c|)$ *in the On-Line Encyclopedia of Integer Sequences (OEIS) [7].*

Because we can compute $|B_\alpha^c|$ from α in linear time (see Lemma 1.4.3), if we use a computational model where factorials are given by an oracle and arithmetic can be computed in constant time, then $\#\text{prefix}_{\mathscr{D}}$ can be computed in linear time with respect to ℓ, the length of the prefix.

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 1.1: 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.

Example 1.4.7. *For example, for $n = 14$, we wish to count the number of derangements that start with the prefix $\alpha = (6,1)$. Since the prefix has two letters, $\ell = 2$ and $n - \ell = 14 - 2 = 12$. Number of squares in B^c_α is*

$$|B^c_\alpha| = 14 - 2 - \underbrace{|\{3,4,\ldots,14\} \cap \{6,1\}|}_{1} = 11 \tag{1.16}$$

Thus there are $A047920(12,11) = 190\,899\,411$ derangements that start with $\alpha = (6,1)$.

Now that we have an efficient algorithm for computing $\#\text{prefix}_{\mathscr{D}} \colon \mathscr{W} \to \mathbb{N}_{\geq 0}$, we can invoke the recursive formula in Lemma 1.2.5 to compute $\text{unrank}_{\mathscr{D}} \colon \mathbb{N}_{\geq 0} \to \mathscr{D}$ and unrank derangements. The sequence of recursive steps is illustrated in Table 1.1.

1.5 Unranking ménage permutations

After claiming a 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 1.5.1. *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 $\mathcal{M}_n$.*

1.5.1 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, offering another prize for unranking ménage permutations. Specifically, he posed the following problem.

\$100 Problem. *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.*

Using the techniques described in this section, we developed a computer program that computed the answer in less than 0.03 seconds. (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.)

\$100 Answer. *The desired permutation is*

$$7\ 16\ 19\ 12\ 2\ 8\ 15\ 1\ 18\ 14\ 3\ 9\ 20\ 10\ 5\ 17\ 13\ 4\ 11\ 6. \tag{1.17}$$

1.5.2 Overview for unranking ménage permutations

As in the section about unranking derangements, we will use the insight from Lemma 1.2.5 that if we can efficiently count the number of words with a give 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 is block diagonal. These complementary blocks have a structure that we can understand, and we can leverage that understanding to compute the rook polynomials of these blocks and consequently of the complementary board itself. Once we have computed the rook polynomial of the complementary board, we can use the principle of inclusion/exclusion to compute the number of full rook placements on the original board. This gives us the number of ménage permutations with a given prefix.

1.5.3 Disjoint board decomposition

Looking at Figure 1.3, it appears that placing rooks according to a prefix 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.

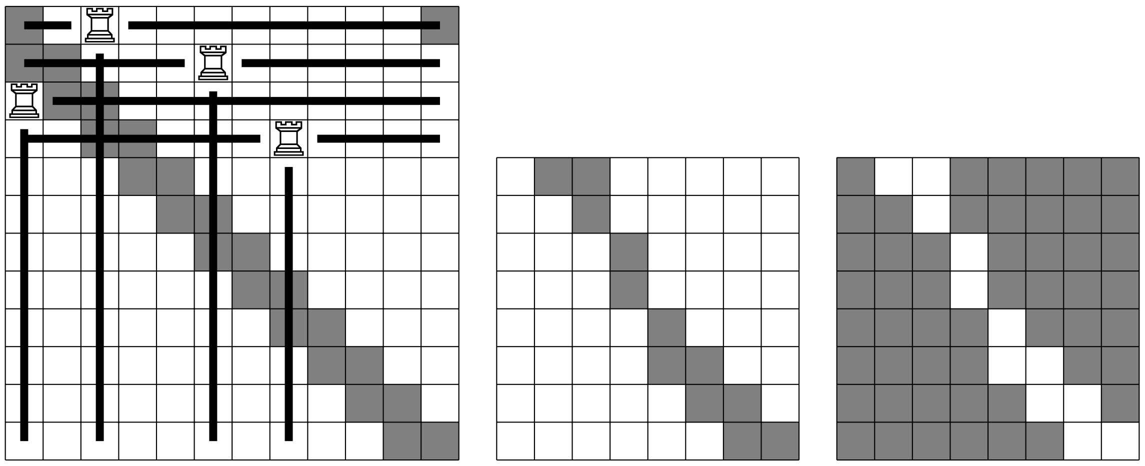

Figure 1.3: The prefix $\alpha = (3,6,1,8)$, the derived board B_α, and the derived complementary board $B_\alpha^c = \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}_7^\intercal$. There are 8062 ways of placing eight nonattacking rooks on B_α.

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 1.5.2. *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 1.5.3 ([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 the b_is*

$$p_B(x) = \prod_{i=1}^{m} p_{b_i}(x). \tag{1.18}$$

We will use this disjoint board decomposition repeatedly, because as Figure 1.3 suggests, 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 1.4.

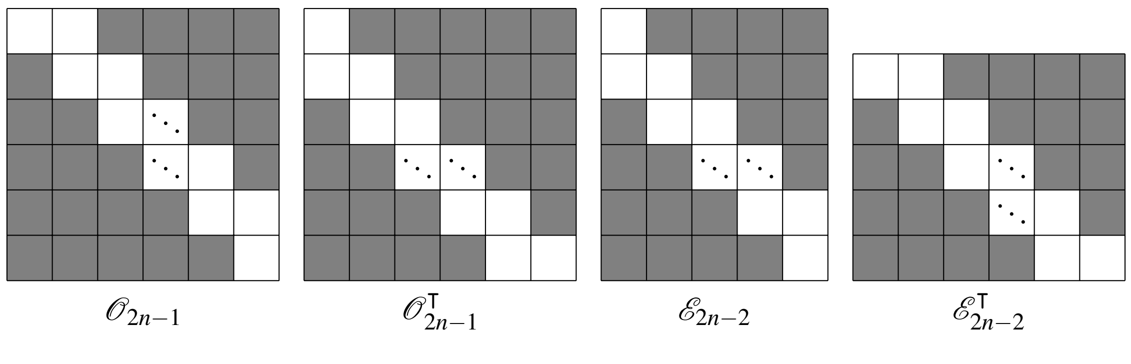
$\mathscr{O}_{2n-1}$
$\mathscr{O}^{\mathsf{T}}_{2n-1}$
$\mathscr{E}_{2n-2}$
$\mathscr{E}^{\mathsf{T}}_{2n-2}$

Figure 1.4: 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 $(n-1) \times n$.

Definition 1.5.4. *A board is called* ***staircase-shaped*** *if it matches one of the following four shapes:*

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

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 1.5.5. *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 complimentary board $B^c_{(\alpha_1)}$ can be partitioned into two disjoint sub-boards with shapes $A_{\alpha_1 - 1}$ and $A^{\mathsf{T}}_{n-\alpha_1}$. (This is illustrated for the case of $n = 7$ and α_1 in Figure 1.5.)

The inductive hypothesis is that the derived complimentary board for a prefix of length $\ell - 1$ consists of sub-boards with shape A_{n_1}, $A^{\mathsf{T}}_{n_2}$, B_{n_3}, or $B^{\mathsf{T}}_{n_4}$. Placing a rook in row ℓ can remove a top row or a column or both in a given sub-board. Table 1.2 below shows the resulting sub-boards after placing a rook in ℓ-th row of B, which may be in the top row or the i-th column of a given sub-board.

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

Table 1.2: 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 results in a board of one of the four described shapes. $\square$

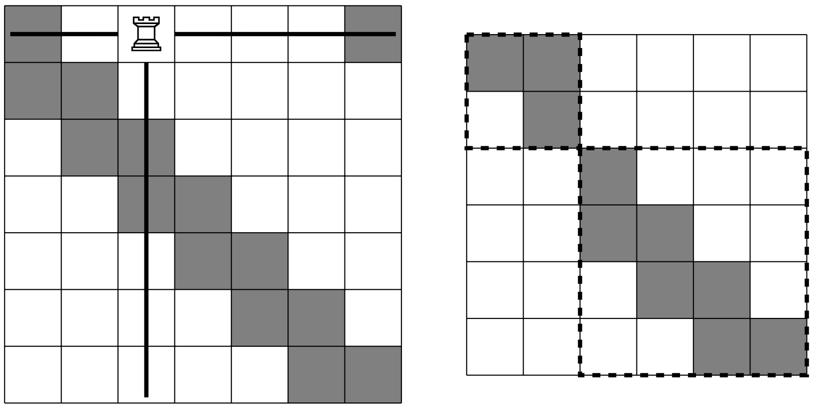
Figure 1.5: The first chessboard shows a placement of a rook at position 3, the second shows how the derived complementary board can be partitioned into two disjoint boards with 3 and 7 squares respectively.

1.5.4 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 easily 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. This family is described by OEIS sequence A011973 [7].

Definition 1.5.6. *For $j \geq 0$, the j-th **Fibonacci polynomial** $F_j(x)$ is defined recursively as:*

$$F_0(x) = 1 \tag{1.19}$$

$$F_1(x) = 1 + x \tag{1.20}$$

$$F_n(x) = xF_{n-2}(x) + F_{n-1}(x). \tag{1.21}$$

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

Lemma 1.5.7. *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 1.3.6, and proceed by induction on the number of squares, always choosing to include or exclude the upper-left square.

Since the reflections of board has the same rook polynomial as the unreflected board, without loss of generality, we will compute the rook polynomials for $\mathscr{O}_{2n-1}$ and $\mathscr{E}_{2n-2}$ 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 rooks on $\mathscr{O}_1$, and no ways to place more than one rook. Thus

$$p_{\mathscr{E}_0}(x) = 1 \qquad = F_0(x), \text{ and} \tag{1.22}$$

$$p_{\mathscr{O}_1}(x) = 1 + x = F_1(x). \tag{1.23}$$

With the base case established, our inductive hypothesis is that $p_B(x) = F_h(x)$ for whenever B is a staircase-shaped boards 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 1.3.6, the rook polynomial of $\mathscr{E}_k$ is

$$p_{\mathscr{E}_k}(x) = x p_{\mathscr{E}_{k-2}}(x) + p_{\mathscr{O}_{k-1}}(x) \tag{1.24}$$

$$= xF_{k-2}(x) + F_{k-1}(x) \tag{1.25}$$

$$= F_k(x). \tag{1.26}$$

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 1.3.6, the rook polynomial of $\mathscr{O}_k$ is

$$\begin{aligned} p_{\mathscr{O}_k}(x) &= xp_{\mathscr{O}_{k-2}}(x) + p_{\mathscr{E}_{k-1}}(x) && (1.27) \\ &= xF_{k-2}(x) + F_{k-1}(x) && (1.28) \\ &= F_k(x). && (1.29) \end{aligned}$$

□

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

Corollary 1.5.8. *Suppose that B^c_α can be partitioned into m disjoint staircase-shaped sub-boards of size $b_1, b_2, \ldots, b_m$. Then the rook polynomial of B^c_α is*

$$p_{B^c_\alpha}(x) = \prod_{i=1}^{m} F_{b_i}, \tag{1.30}$$

where F_j is the j-th Fibonacci polynomial.

Proof. This follows directly Theorem 1.5.3 together with Lemma 1.5.7. □

1.5.5 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 1.5.9. *Given a nonempty prefix $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ 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:*

$$c_i = \begin{cases} 0 & i < \ell \\ 1 & i = \ell \text{ or } i = n \\ 2 & \ell < i < n \end{cases} \tag{1.31}$$

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 $(1,n)$:

$$B^c = \{(i,i) : i \in [n]\} \cup \{(i+1,i) : i \in [n-1]\} \cup \{(1,n)\}. \tag{1.32}$$

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

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

$$P_\alpha = \left\{ \sum_{p \in \mathcal{P}_i} c_p : \mathcal{P}_i \in \mathcal{P} \right\}. \tag{1.33}$$

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 for 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 1.5.11. *As illustrated in Figure 1.3, if $n = 12$ and $\alpha = (3,6,1,8)$, then the contiguous partition of*

$$[12] \setminus \{3,6,1,8\} = \{\underbrace{2}_{P_1}, \underbrace{4,5}_{P_2}, \underbrace{7}_{P_3}, \underbrace{9,10,11,12}_{P_4}\} \tag{1.34}$$

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:

$$B_\alpha^c = \mathscr{E}_0 \sqcup \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}_7^\intercal, \tag{1.35}$$

1.5.6 Complementary polynomials to ménage permutations with a given prefix

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 1.4.5, this allows us can 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 1.5.12. *The number of ménage permutations that begin with a valid, nonempty prefix α is*

$$\#\text{prefix}_{\mathcal{M}}(\alpha) = \sum_{k=0}^{n} (-1)^k r_k^c (n-k)! \tag{1.36}$$

where $\sum_{k=0}^{n} r_k^c x^k = \prod_{p \in P_\alpha} F_p$, F_k is the k-th Fibonacci polynomial, and P_α 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 1.5.8 together with Lemma 1.5.10 □

This together with Lemma 1.2.5 allows an efficient unranking algorithm.

Now that we have all of the ingredients to unrank menage permutations, we can give a full example that computes the number of ménage permutations with a given prefix.

Example 1.5.13. *We will continue with the running example illustrated in Figure 1.3 and expounded on in Example 1.5.11.*

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

$$B_\alpha^c = \mathscr{E}_0 \sqcup \mathscr{O}_3 \sqcup \mathscr{E}_2 \sqcup \mathscr{O}_7^{\intercal}. \tag{1.37}$$

Lemma 1.5.7 tells us that the rook polynomial of B_α^c is

$$\begin{aligned}
p_{B_\alpha^c}(x) &= F_3(x)F_2(x)F_7(x) && (1.38)\\
&= (1+3x+x^2)(1+2x)(1+7x+15x^2+10x^3+x^4) && (1.39)\\
&= 1+12x+57x^2+136x^3+170x^4+105x^5+27x^6+2x^7 && (1.40)\\
&= \sum_{k=0}^{7} r_k^c x^k. && (1.41)
\end{aligned}$$

By Lemma 1.4.5, the number of ways to place eight rooks on B_α is is

$$\begin{aligned}
N_0 &= \sum_{k=0}^{7} (-1)^k r_k^c (8-k)! && (1.42)\\
&= 1(8!) - 12(7!) + 57(6!) - 136(5!) + 170(4!) - 105(3!) + 27(2!) - 2(1!) && (1.43)\\
&= 8062. && (1.44)
\end{aligned}$$

α	$\#\text{prefix}(\alpha)$	index range	block sizes	$\text{unrank}_{\mathcal{M}}(i)$
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	$(946,970]$	$(0,2,5)$	$f_{1000}^{\mathcal{M}}(351,946)$
$\cdots$	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	$(975,980]$	$(0,5)$	$f_{1000}^{\mathcal{M}}(3542,975)$
$\cdots$	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)	$f_{1000}^{\mathcal{M}}(354821,999)$
$\cdots$	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 1.3: The recursive computation of the 1000th ménage permutation.

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 Lemma 1.2.5 to unrank derangements.

Example 1.5.14. *There are* $A000179(8) = 4738$ *ménage permutations on* 8 *letters. Table 1.3 shows the steps of the algorithm that determines that the* 1000*th ménage permutation in lexicographic order is*

$$w_{1000} = 3\ 5\ 4\ 8\ 2\ 7\ 1\ 6. \tag{1.45}$$

1.6 Generalizations and open questions

There are a number of different directions and potential generalizations of unranking problems for derangements and ménage permutations. A partial list of further directions is unranking other restricted permutations (e.g. discordant permutations), unranking other kinds of restrictions on words (e.g. Lyndon words), unranking other kinds of combinatorial objects (e.g. unlabeled graphs), or unranking permutations with a different underlying total order.

1.6.1 Other restricted permutations

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

Definition 1.6.1 ([8]). *Let $S \subset \mathbb{Z}$ be a finite collection of integers. A S-**avoiding permutation** is a permutation $\pi \in S_n$ such that*

$$\pi(i) - i - s \not\equiv 0 \bmod n \textit{ for all } i \in [n] \textit{ and } s \in S. \tag{1.46}$$

Example 1.6.2. *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 generalize quite easily to the case of $\{i\}$-avoiding and $\{i, i+1\}$ avoiding permutations.

Open Question 1.6.3. *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. As such, we are interested in a more specific question.

Open Question 1.6.4. *Do there exist efficient unranking algorithms on* $\{-1,1\}$*-avoiding permutations?*

The main obstruction to using the techniques from Section 1.5 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.

1.6.2 Permutation statistics

Another area for exploration, inspired by Chapter **??**, is deranking permutations with a given permutation statistic.

Open Question 1.6.5. *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

$$\mathscr{I}_n^k = \{\pi \in S_n \,|\, \operatorname{inv}(\pi) = k\}, \tag{1.47}$$

and if so, how does one construct it?

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

1.6.3 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.

Example 1.6.6. *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 1.6.7. *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?*

1.6.4 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 [9] 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 1.6.8. *A* ***Lyndon word*** *is a string over an alphabet of letters that is the unique minimum with respect to all of its rotations.*

Example 1.6.9. 00101 *is a Lyndon word because*

$$00101 = \min\{00101, 01010, 10100, 01001, 10010\} \tag{1.48}$$

is the unique minimum of all of its rotations.

011011 *is not a Lyndon word because while*

$$011011 = \min\{011011, 110110, 101101, 011011, 110110, 101101\}, \tag{1.49}$$

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

Definition 1.6.10 ([10])**.** *Suppose that $\{a_i\}_{i=1}^{\infty}$ and $\{b_i\}_{i=1}^{\infty}$ are integer sequences related by*

$$1 + \sum_{n=1}^{\infty} b_n x^n = \prod_{i=1}^{\infty} \frac{1}{(1 - x^i)^{a_i}}. \tag{1.50}$$

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 1.6.11. *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 1.6.12. *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 1.4.

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

Conjecture 1.6.13. *For $k \geq 1$ Let $\mathscr{L}_\alpha$ be the sequence of the number of Lyndon words of length n with prefix $\alpha = (0,0,\dots,0)$ 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 1.6.14. *For $k \geq 2$ Let $\mathscr{L}_\alpha$ be the sequence of the number of Lyndon words of length n with prefix $\alpha = (1,0,0,\dots,0)$ 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, for all prefixes that are not the zero sequence.

Conjecture 1.6.15. *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 1.6.16. *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?*

α	$\mathscr{L}_\alpha$ and $\mathcal{E}(\mathscr{L}_\alpha)$												Conjectured recurrence	
0	1	1	2	3	6	9	18	30	56	99	186	335	$a_{n+1} = 2a_n$	$n \geq 2$
	1	1	2	4	8	16	32	64	128	256	512	1024		
00	0	0	1	2	4	7	14	25	48	88	168	310	$a_{n+1} = 2a_n$	$n \geq 4$
	1	0	0	1	2	4	8	16	32	64	128	256		
01	0	1	1	1	2	2	4	5	8	11	18	25	$a_{n+2} = a_{n+1} + a_n$	$n \geq 1$
	1	0	1	1	2	3	5	8	13	21	34	55		
000	0	0	0	1	2	4	8	15	30	57	112	214	$a_{n+1} = 2a_n$	$n \geq 5$
	1	0	0	0	1	2	4	8	16	32	64	128		
001	0	0	1	1	2	3	6	10	18	31	56	96	$a_{n+3} = a_{n+2} + a_{n+1} + a_n$	$n \geq 1$
	1	0	0	1	1	2	4	7	13	24	44	81		
010	0	0	0	0	1	1	2	3	5	7	12	18	$a_{n+2} = a_{n+1} + a_n$	$n \geq 5$
	1	0	0	0	0	1	1	2	3	5	8	13		
011	0	0	1	1	1	1	2	2	3	4	6	7	$a_{n+3} = a_{n+2} + a_n$	$n \geq 1$
	1	0	0	1	1	1	2	3	4	6	9	13		
0000	0	0	0	0	1	2	4	8	16	31	62	121	$a_{n+1} = 2a_n$	$n \geq 6$
	1	0	0	0	0	1	2	4	8	16	32	64		
0001	0	0	0	1	1	2	4	7	14	26	50	93	$a_{n+4} = a_{n+3} + a_{n+2} + a_{n+1} + a_n$	$n \geq 1$
	1	0	0	0	1	1	2	4	8	15	29	56		
0010	0	0	0	0	1	1	3	5	9	16	30	53	$a_{n+3} = a_{n+2} + a_{n+1} + a_n$	$n \geq 6$
	1	0	0	0	0	1	1	3	5	9	17	31		
0011	0	0	0	1	1	2	3	5	9	15	26	43	$a_{n+4} = a_{n+3} + a_{n+2} + a_n$	$n \geq 1$
	1	0	0	0	1	1	2	3	6	10	18	31		
0101	0	0	0	0	1	1	2	3	5	7	12	18	$a_{n+2} = a_{n+1} + a_n$	$n \geq 5$
	1	0	0	0	0	1	1	2	3	5	8	13		
0110	0	0	0	0	0	0	1	1	1	2	3	4	$a_{n+3} = a_{n+2} + a_n$	$n \geq 6$
	1	0	0	0	0	0	0	1	1	1	2	3		
0111	0	0	0	1	1	1	1	1	2	2	3	3	$a_{n+4} = a_{n+3} + a_n$	$n \geq 1$
	1	0	0	0	1	1	1	1	2	3	4	5		

Table 1.4: A table of conjectures about the number of Lyndon words of length n and prefix α. Each row contains a prefix α a sequence counting Lyndon words with α ($\mathscr{L}_\alpha$), the Euler transformation of that sequence ($\mathcal{E}(\mathscr{L}_\alpha)$), a conjectured recurrence for the Euler-transformed sequence, and the valid range for the conjectured recurrence.