Flip-sort and combinatorial aspects of pop-stack sorting

Andrei Asinowski[1] Cyril Banderier[2] Benjamin Hackl[1]

[1] *University of Klagenfurt, Klagenfurt, Austria*
[2] *University of Sorbonne Paris Nord, Villetaneuse, France*

Submitted 9th Mar. 2020.

Flip-sort is a natural sorting procedure which raises fascinating combinatorial questions. It finds its roots in the seminal work of Knuth on stack-based sorting algorithms and leads to many links with permutation patterns. We present several structural, enumerative, and algorithmic results on permutations that need few (resp. many) iterations of this procedure to be sorted. In particular, we give the shape of the permutations after one iteration, and characterize several families of permutations related to the best and worst cases of flip-sort. En passant, we also give some links between pop-stack sorting, automata, and lattice paths, and introduce several tactics of bijective proofs which have their own interest.

Keywords: permutation, stack sorting, generating function, finite automaton, generating tree, lattice path

1 Introduction and definitions

1.1 Flip-sort and pop-stack sorting

Sorting algorithms are addressing one of the most fundamental tasks in computer science, and, accordingly, have been intensively studied. This is well illustrated by the third volume of Knuth's *Art of Computer Programming* [22], which tackles many questions related to the worst/best/average case behaviour of sorting algorithms, and gives many examples of links between these questions and the combinatorial structures hidden behind permutations (see also [11, Chapter 8.2] and [27]). For example, in [21, Sec. 2.2.1], Knuth considers the model of permutations sortable by one stack (neatly illustrated by a sequence of n wagons on a railway line, with a side track that can be used for permuting wagons) and show that they are characterized as 231-avoidable permutations: this result is a cornerstone of the field of permutation patterns. Since that time, many results were also obtained e.g. on permutations sortable with 2 stacks, which offer many nice links with the world of planar maps [15, 31]. The analysis of sorting with 3 stacks remains an open problem. Numerous variants were considered (stacks in series, in parallel, etc.). Our article pursues this tradition by dealing with the combinatorial aspects of the flip-sort algorithm, a procedure for sorting permutations via a **pop-stack** (as considered in [6, 7]), which we now detail.

We use the following notation. Permutations will be written in one-line notation, $\pi = a_1 a_2 \dots a_n$. An *ascent* (resp. *descent*) is a gap between two neighbouring positions, i and $i+1$, such that $a_i < a_{i+1}$ (resp. $a_i > a_{i+1}$). Descents of π split it into *runs* (maximal ascending strings), and ascents split it into *falls* (maximal descending strings). For example, the permutation 3276145 splits into runs as 3|27|6|145, and into falls as 32|761|4|5.

Flip-sort consists into iterating flips on the input permutation, where *flip* is the transformation T that reverses *all* the falls of a permutation. For example, $T(3276145) = 2316745$. If one applies T repeatedly (thus obtaining $T(\pi), T^2(\pi), \dots$), then one eventually obtains the identity permutation id. For example: $3276145 \mapsto 2316745 \mapsto 2136475 \mapsto 1234657 \mapsto 1234567$; see Figure 1 for a visualization of this example by means of permutation diagrams. Nota bene: if one does not impose the stack to contain increasing values, then the process can be related to the famous pancake sorting problem (see [18]).

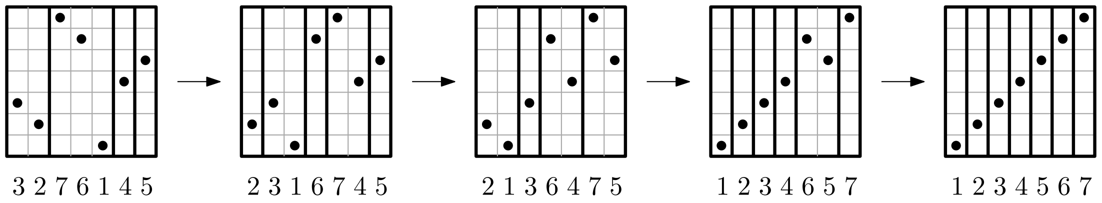


Fig. 1: The flip-sort algorithm iteratively flips all the falls until the permutation gets sorted. In these drawings, the solid vertical bars separate the falls. Each iteration can also be seen as sending the input into a pop-stack: while reading the permutation from left to right, each decreasing value is added to the stack, and *the whole content* of the stack is popped when an increasing value is met.

Let the *cost* of π be the (minimum) number of iterations needed to sort π, that is, $\mathsf{cost}(\pi) = \min\{m \geq 0\colon\ T^m(\pi) = \mathsf{id}\}$. To see that the cost is always finite, observe that at each iteration the number of inversions strictly decreases until id is reached. It follows that, for any permutation of size n, we have $\mathsf{cost}(\pi) \leq \binom{n}{2}$. In fact, the cost is always lower than this naive upper bound: Ungar [30] proved[(i)] that one always has $\mathsf{cost}(\pi) \leq n-1$ (we refine this result in our Section 4.2). Moreover, this bound is tight since for each n there are permutations of size n with cost $n-1$ (we give some examples in Section 4.4).

The pop-stack sorting leads to many interesting combinatorial questions. Some of them can be visualized by means of the *pop-stack-sorting tree* whose nodes are all permutations of size n, with directed arcs from π to $T(\pi)$. It is convenient to draw such a tree with levels corresponding to the distance to id: see Figure 2 that shows the tree for $n = 5$ (the arcs are understood to be directed from left to right). When, more generally, one considers this tree for any value of n, it is natural to ask for a characterization of the permutations that need few iterations of T to be sorted (i.e., the right side of the tree), or many iterations (i.e., the left side of the tree); or the image of T (i.e., the non-leaves of the tree). We address these questions in this article, thus extending previous works [13, 26, 30].

Another challenging problem is describing a typical shape of $T^m(\pi)$, as m runs from 0 to $n-1$, for random π. Computer simulations show a very distinctive shape of the corresponding permutation diagrams: they possess some empty areas and a cloud of points accumulating along two curves that tend to unite and ultimately converge to the identity permutation; see Figure 3. We prove a more rigorous formalization of some of these claims in this article.

1.2 Summary of our results

The article is organized as follows.

In Section 2, we study the permutations that belong to $\mathrm{Im}(T)$, the **image** of the flip transformation. We find a structural characterization of such permutations, and we prove that the generating function of these permutations is rational if the number of runs is fixed. We then use a generating tree approach to design some efficient algorithms enumerating such permutations. We also show that their generating function satisfies a curious functional equation, and we give an asymptotic bound.

Section 3 is dedicated to permutations with **low cost**. A permutation π is *k-pop-stack-sortable* if $\mathsf{cost}(\pi) \leq k$, or, equivalently, $T^k(\pi) = \mathsf{id}$. Avis and Newborn showed that 1-pop-stack-sortable are precisely the layered permutations [7], while k-pop-stack-sortable permutations are recognizable by an automaton (see Claesson and Guðmundsson [13]). For $k = 2$, Pudwell and Smith [26] listed some conjectures that we prove via a bijection between 2-pop-stack-sortable permutations to a family of lattice paths.

In contrast, in Section 4, we deal with permutations with **high cost**, and with $\mathrm{Im}(T^m)$ for arbitrary m. Our main result is a (tight) bound on **bandwidth** of $\tau \in \mathrm{Im}(T^m)$, which provides a partial explanation of the phenomena that we observe in Figure 3. Additionally, we find a full characterization of $\mathrm{Im}(T^{n-2})$, and some conditions for $\mathsf{cost}(\tau) = n-1$. We conclude with a conjecture concerning the cost of skew-layered permutations.

[(i)] This result was conjectured by Goodman and Pollack [19], and proved by Ungar, in the context of *allowable sequences of permutations* in links with the minimum number of directions determined by n non-collinear points in the plane; see the crisp presentation of this nice geometrical question in [1, Chapter 12].

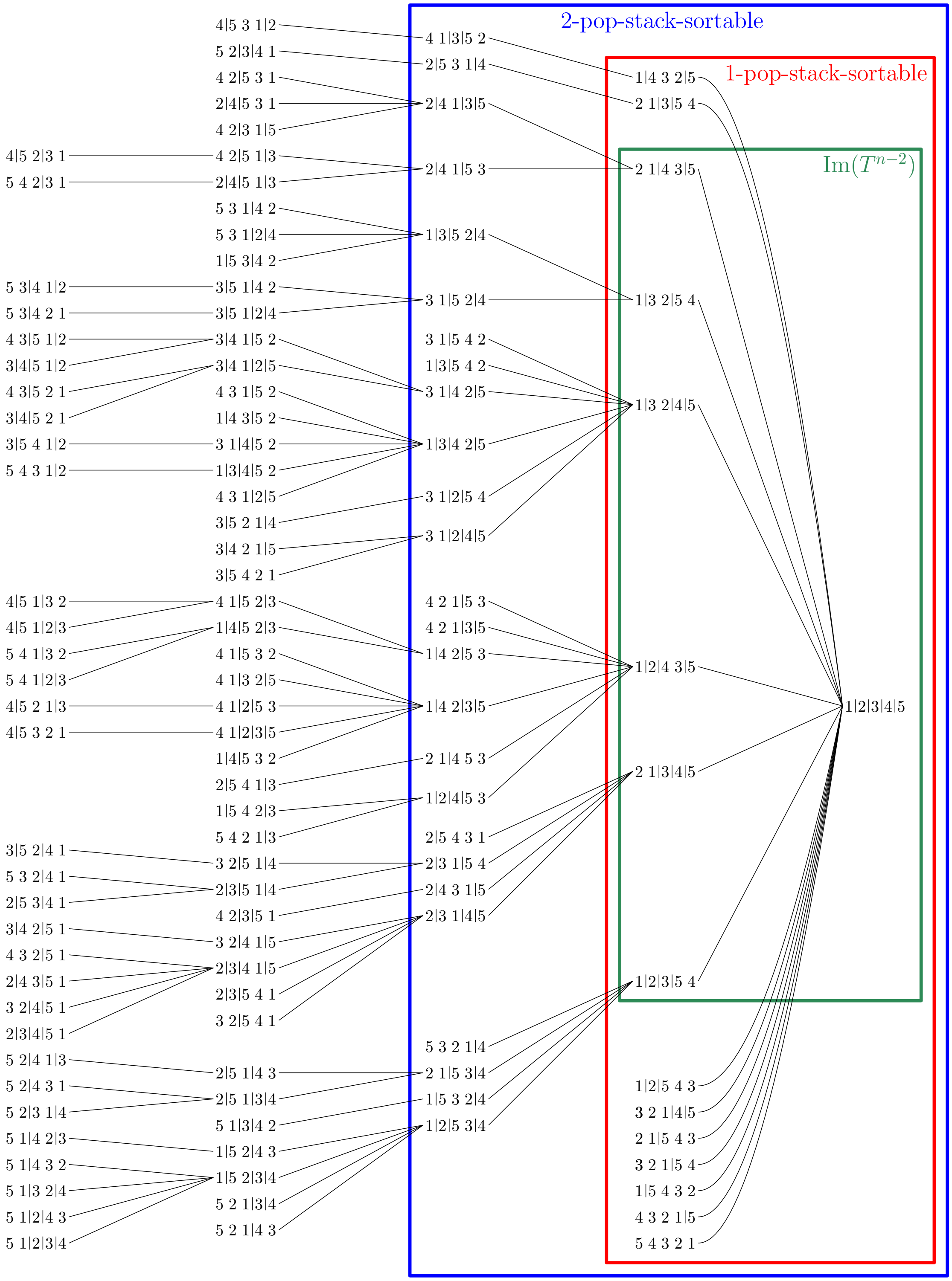


Fig. 2: The pop-stack-sorting tree for $n = 5$. Flipping the falls in each permutation leads to its successor until one reaches the identity permutation. The leftmost column thus corresponds to the worst cases of this procedure. In this figure, we also mark three sets considered in this article: 1- and 2-pop-stack-sortable permutations, and Im(T^{n-2}). Additionally, Im(T) consists of all internal nodes.

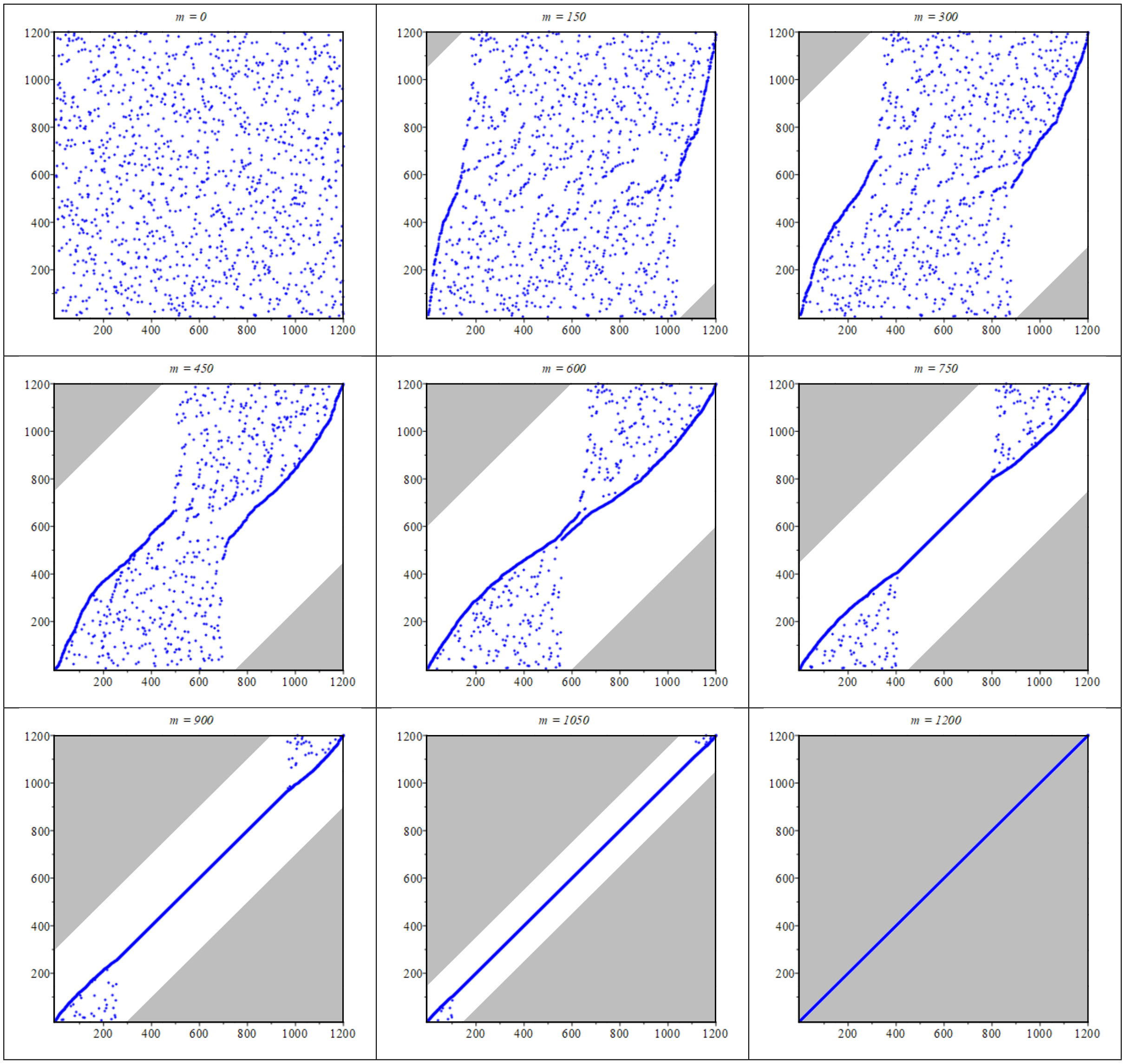


Fig. 3: The evolution of the flip-sort algorithm on a sample permutation of size $n = 1200$. In this article, we show that for any input permutation, there are two areas (the grey areas in the above plots, which are proportional to the number m of iterations) without any dots inside. This entails that the permutations "shrink" (in the white area) until they finally get fully sorted (after $\leq n - 1$ iterations). We also show that, for all m, for any coordinates in the white area, there exists an input permutation which will have a dot at these coordinates.

2 Results concerning one iteration of the flip-sort algorithm

2.1 Structural characterization of pop-stacked permutations

We start this section with a characterization of the image of T – internal nodes in the pop-stack tree from Figure 2)

Definition 1. A permutation τ is *pop-stacked* if it belongs to $\mathrm{Im}(T)$, that is, if there is a permutation π such that we have $\tau = T(\pi)$.

Definition 2. A pair (R_i, R_{i+1}) of adjacent runs of π is overlapping if $\min(R_i) < \max(R_{i+1})$. (Notice that $\max(R_i) > \min(R_{i+1})$ holds automatically since R_i and R_{i+1} are *distinct* runs.)

We begin our investigations of the image of T by giving a characterization of the permutations in $\mathrm{Im}(T)$ in terms of overlapping runs. In fact, in the following theorem we prove that a permutation is pop-stacked if and only if all pairs of adjacent runs are overlapping. See Figure 4 for a schematic drawing that represents the structure of permutations with overlapping adjacent runs, and an example. Despite its very natural definition, this family of permutations was, to the best of our knowledge, never studied before our initial conference contributions [3,4].

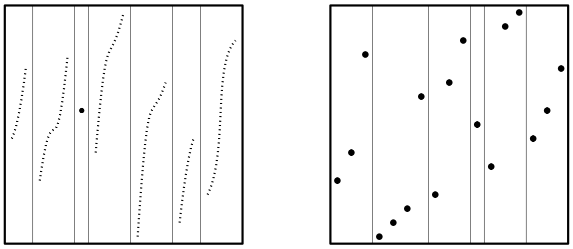


Fig. 4: Pop-stacked permutations: a general schematic drawing of permutation with adjacent overlapping runs (a run can be of length 1), and an example: the permutation 5 7 14 | 1 2 3 11 | 4 12 15 | 9 | 6 16 17 | 8 10 13.

Theorem 3. *A permutation τ is pop-stacked if and only if all its pairs (R_i, R_{i+1}) of adjacent runs are overlapping.*

Proof: [First part of the proof: pop-stacked ⇒ overlapping runs.] Let $\tau = b_1 b_2 \dots b_n$ be a permutation with $\min(R_i) > \max(R_{i+1})$ for some pair (R_i, R_{i+1}) of its adjacent runs. Let b_α be the last letter in R_i and $b_{\alpha+1}$ be the first letter in R_{i+1} (that is, $R_i = [\dots, b_\alpha]$, $R_{i+1} = [b_{\alpha+1}, \dots]$).

Assume for contradiction that we have $\tau = T(\pi)$ for some permutation $\pi = a_1 a_2 \dots a_n$. In π, we have $a_\alpha < a_{\alpha+1}$ because otherwise the string $[a_\alpha a_{\alpha+1}]$ is a part of a fall in π, and upon applying T we have $b_\alpha < b_{\alpha+1}$, which is impossible, because b_α and $b_{\alpha+1}$ lie in different runs of τ. Therefore, if we consider the partition of π into falls, then a_α is the last letter of some fall F_j, and $a_{\alpha+1}$ is the first letter of the next fall F_{j+1}. However, the values of F_j are a subset of those of R_i, and the values of F_{j+1} are a the subset of those of R_{i+1}. Therefore we have $a_\alpha > a_{\alpha+1}$, which is a contradiction to $a_\alpha < a_{\alpha+1}$ observed above.

[Second part of the proof: overlapping runs ⇒ pop-stacked.] Consider a permutation τ with $\min(R_i) < \max(R_{i+1})$ for all pairs (R_i, R_{i+1}) of adjacent runs. Let π be the permutation obtained from τ by reversal of all its runs. Then the partition of π into falls is the same as the partition of τ into runs, and, therefore, π is a (not necessarily unique) pre-image of τ. □

2.2 Pop-stacked permutations with a fixed number of runs

Sections 2.2–2.4 are dedicated to enumerative aspects of pop-stacked permutations. Let p_n denote the number of pop-stacked permutations of size n. The sequence $(p_n)_{n\geq 1}$ starts with 1, 1, 3, 11, 49, 263, 1653, 11877, 95991, 862047, 8516221, 91782159, 1071601285, 13473914281, 181517350571, 2608383775171, 39824825088809, 643813226048935; we have added it as A307030 to the On-Line Encyclopedia of Integer Sequences. While it is hard to compute more terms directly, the introduction of additional parameters provides further insights. Specifically, in this section we consider the number of runs in pop-stacked permutations. In particular, we show that for each fixed k, the generating function for pop-stacked permutations with exactly k runs is rational.

For fixed k, let $p_{n,k}$ denote the number of pop-stacked permutations of size n with exactly k runs. The case $k = 1$ is trivial: for any size, the only permutation with only one run is the identity permutation, and it is pop-stacked as it is e.g. the image of itself. Thus, we have $p_{n,1} = 1$ for each $n \geq 1$. Note that $p_{n,k}$ for $k > 1$ is always an even number (indeed, listing the runs from the last one to the first one gives an involution among pop-stacked permutations), therefore p_n is always an odd number.

One key ingredient of our further results is the following encoding of permutations, which we call *scanline mapping*. Let π be a permutation with k runs. Let r_i be the index of the run in which the letter i lies. Consider the word $w(\pi) = r_{\pi^{-1}(1)}\, r_{\pi^{-1}(2)} \, \cdots \, r_{\pi^{-1}(n)} \in \{1, \ldots, k\}^n$. Visually, we scan the graph of π from the bottom to the top, and for each point that we encounter in this order, we record to which run it belongs; see Figure 5.

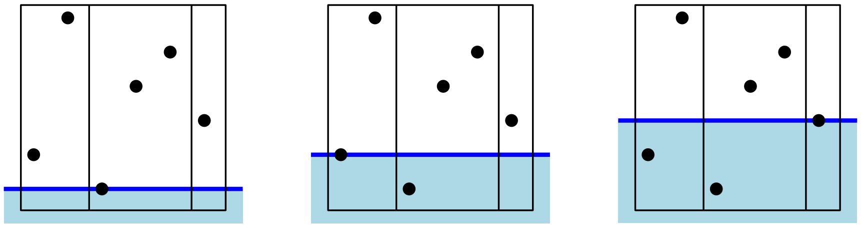


Fig. 5: The *scanline mapping* allows us to encode permutations by words: above, the permutation $\pi = 261453$ is encoded by the word $w(\pi) = 213221$ (the sequence of runs encountered if one reads the permutation bottom-to-top).

Proposition 4. 1. *The scanline mapping $\pi \mapsto w(\pi)$ is injective.*

2. *The scanline mapping $\pi \mapsto w(\pi)$ induces a bijection between permutations of size n with k runs* and *the sequences in $\{1, \ldots, k\}^n$ in which there is an occurrence of $j + 1$ before an occurrence of j, for each j $(1 \leq j \leq n - 1)$.*

3. *The scanline mapping $\pi \mapsto w(\pi)$ induces a bijection between pop-stacked permutations of size n with k runs* and *the sequences in $\{1, \ldots, k\}^n$ in which there is an occurrence of $j + 1$ before an occurrence of j, and also an occurrence of j before an occurrence of $j + 1$, for each j $(1 \leq j \leq n - 1)$.*

Proof:

1. The positions of i ($1 \leq i \leq k$) in $w(\pi)$ are the values in the ith run of π. Thus, π is reconstructed from $w(\pi)$ uniquely.

2. If (and only if) for some j, all the occurrences of j in $w(\pi)$ are before the occurrences of $j+1$, then the corresponding positions of π do not form two distinct runs and thus we do not get a permutation with k runs.

3. If (and only if) for some j, all the occurrences of $j+1$ in $w(\pi)$ are before the occurrences of j, then all the values in the jth run of π are larger than all the values in the $(j+1)$st run, and thus these runs are not overlapping. □

Proposition 4 can be used to obtain a formula for the case of two runs directly:

Proposition 5. *For $n \geq 1$, the number of pop-stacked permutations of size n with exactly two runs, is $p_{n,2} = 2^n - 2n$.*

Proof: By Proposition 4, $p_{n,2}$ is the number of sequences in $\{1,2\}^n$ with an occurrence of 1 before an occurrence of 2, and also an occurrence of 2 before an occurrence of 1. There are $2n$ sequences that violate this condition (including the "all-1" and the "all-2" sequences). □

It is pleasant to have a combinatorial explanation for the number of pop-stacked permutations with two runs, and this game could be pursued (algorithmically) for a fixed number of runs k, but to get a closed-form formula holding for any arbitrary number of runs is still open. While the following theorem does not give an explicit formula for all of those cases immediately, it proves that the counting sequence for pop-stacked permutations with precisely k runs is nice and simple from a structural point of view.

Theorem 6. *Let $k \geq 1$ be fixed. Then $P_k(z) := \sum_{n\geq 1} p_{n,k} z^n$, the generating function for the number of pop-stacked permutations with exactly k runs, is rational.*

Proof: It is well known that the generating function of words recognized by an automaton is rational (see e.g. [16, Sec. I.4.2]). We use Proposition 4 to construct a deterministic automaton $\mathcal{A}_k$ over the alphabet $\{1,\dots,k\}$ that precisely recognizes the words $w(\pi)$ that correspond to the pop-stacked permutations. The states of $\mathcal{A}_k$ are labelled by pairs (L, C), where

- $L \subseteq \{1,\dots,k\}$ indicates the already visited letters, and
- $C \subseteq \bigcup_{j=1}^{k-1} \{(j, j+1), (j+1, j)\}$ indicates the already fulfilled conditions "there is an occurrence of j before an occurrence of $j+1$" resp. "there is an occurrence of $j+1$ before an occurrence of j", such that
- if $j, j+1 \in L$, then at least one of $(j, j+1)$ and $(j+1, j)$ belongs to C.

It is then straightforward to see that $\mathcal{A}_k$ precisely recognizes those words in $\{1,2,\dots,k\}^n$ that correspond bijectively to the pop-stacked permutations of size n with k runs by Proposition 4. Figure 6 shows such an automaton for $k = 3$. □

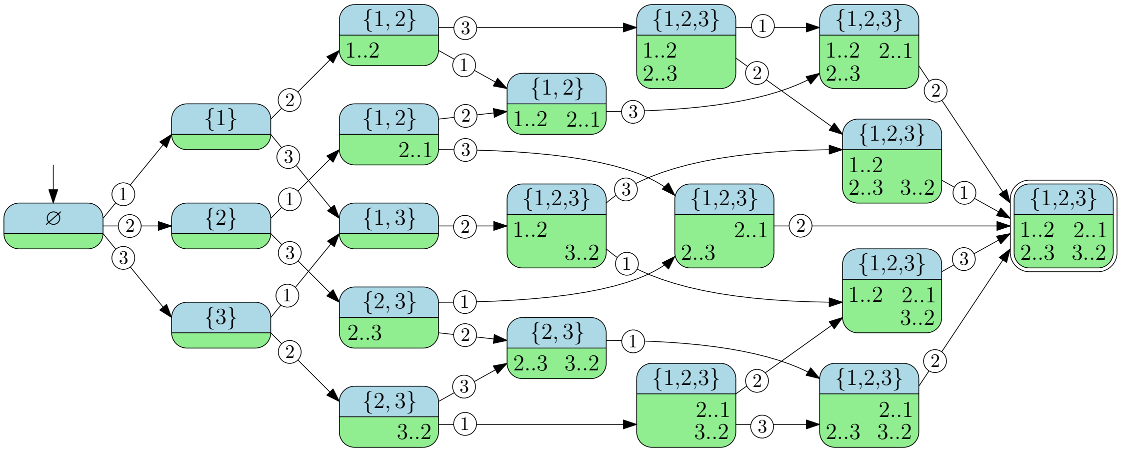


Fig. 6: Automaton $\mathcal{A}_3$ that recognizes the words encoding the pop-stacked permutations with $k = 3$ runs. In each state, the upper blue area indicates the letters from the alphabet $\{1, \ldots, k\}$ already encountered, and the lower green area contains $i..j$ (where $j = i \pm 1$) if and only if there was already an occurrence of i before an occurrence of j in the word read by this point. If a state is entered after reading the letter j, then reading j again loops back to the same state: such transitions are omitted in this figure.

In the next theorem we address the complexity of $\mathcal{A}_k$: its number of states grows roughly as 3.41^k. This exponential growth of the number of states also gives an insight on the complexity of the generating functions associated to these automata.

Theorem 7. *Denote by a_k the number of states in the automaton $\mathcal{A}_k$. The sequence $(a_k)_{k\geq 1}$ satisfies the linear recurrence $a_k = 4a_{k-1} - 2a_{k-2}$. Together with initial conditions, this implies that $(a_k)_{k\geq 1}$ is* A006012*: $(2, 6, 20, 68, 232, \ldots)$. Accordingly, the number of states of $\mathcal{A}_k$ grows exponentially: $a_k = \Theta((2 + \sqrt{2})^k)$.*

Proof: We proceed by induction on k (starting at $k \geq 3$). Denote by Q_k the set of states of the automaton $\mathcal{A}_k$ defined in the proof of Theorem 6. Recall that the states are labelled by (L, C), where L is the set of already encountered letters, and C is the list of already fulfilled conditions of the kind $(j, j + 1)$ or $(j + 1, j)$.

We partition the states of $\mathcal{A}_k$ into three parts as follows.

1. The states of $\mathcal{A}_k$ whose letters do not contain k: they are precisely all the states of $\mathcal{A}_{k-1}$.
2. The states of $\mathcal{A}_k$ whose letters contain k but do not contain $k - 1$. They correspond bijectively to the states of $\mathcal{A}_{k-1}$ whose letters do not contain $k - 1$, and thus to all the states of $\mathcal{A}_{k-2}$:

$$(L, C) \in Q_{k-2} \longleftrightarrow (L, C) \in Q_{k-1} \longleftrightarrow (L \cup \{k\}, C) \in Q_k, \tag{1}$$

   where $k - 1 \notin L$.

3. Finally, the states of $\mathcal{A}_k$ whose letters contain both k and $k-1$. They correspond $(3:1)$-bijectively to the states of $\mathcal{A}_{k-1}$ whose letters contain $k-1$:
$$(L, C) \in Q_{k-1} \quad \longleftrightarrow \quad \left.\begin{array}{l}(L \cup \{k\}, C \cup \{(k-1,k)\}),\\ (L \cup \{k\}, C \cup \{(k,k-1)\}),\\ (L \cup \{k\}, C \cup \{(k-1,k),(k,k-1)\})\end{array}\right\} \in Q_k, \tag{2}$$
where $k-1 \in L$.

Therefore, summing over these three cases, we have
$$a_k = a_{k-1} + a_{k-2} + 3(a_{k-1} - a_{k-2}), \qquad \text{and thus } a_k = \frac{(2+\sqrt{2})^k - (2-\sqrt{2})^k}{\sqrt{2}}, \tag{3}$$
which completes the proof. □

Remark: Upon performing *minimization* on $\mathcal{A}_k$, we obtain automata with the number of states given by $(b_k)_{k\geq 1} = (2, 6, 16, 40, 98, \dots)$. This leads us to the conjecture that this sequence satisfies the linear recurrence $b_k = 3b_{k-1} - b_{k-2} - b_{k-3}$, and that it is in fact A293004[(ii)]. Possibly, this could be proved by following the impact of each step of Brzozowski's algorithm for minimizing an automaton, see [12]. It is interesting to notice that the exponential growth of the number of states would then drop from $a_k = \Theta((2+\sqrt{2})^k)$ to $b_k = \Theta((1+\sqrt{2})^k)$.

Before we continue with our investigations towards efficient computation of these numbers, we want to make some observations concerning the particular shape of the rational generating functions $P_k(z)$. Recall that Theorem 6 provides a construction for a deterministic finite automaton that recognizes pop-stacked permutations of length n. Given such an automaton, the associated generating function can be extracted in a straightforward way. Doing so for $1 \leq k \leq 5$ yields the rational functions
$$\begin{aligned}
P_1(z) &= \frac{z}{1-z},\\
P_2(z) &= \frac{2z^3}{(1-z)^2(1-2z)},\\
P_3(z) &= \frac{2z^4(1+3z-6z^2)}{(1-z)^3(1-2z)^2(1-3z)},\\
P_4(z) &= \frac{2z^6(21-74z+5z^2+180z^3-144z^4)}{(1-z)^4(1-2z)^3(1-3z)^2(1-4z)},\\
P_5(z) &= \frac{2z^7(21+198z-3856z^2+18982z^3-40581z^4+33060z^5+12784z^6-37600z^7+17280z^8)}{(1-z)^5(1-2z)^4(1-3z)^3(1-4z)^2(1-5z)}.
\end{aligned}$$

[(ii)] The goddess of combinatorics is thumbing her nose at us, as this sequence is also related to permutation patterns in many ways: it counts permutations related to the elevator problem [22, 5.4.8 Ex.8], permutations avoiding 2413, 3142, 2143, 3412, and some guillotine partitions [5]!

Some further functions can still be computed in reasonable time. However, as the number of states in the automaton grows exponentially in k this approach is not feasible for large values of k.

Note that the lowest degree in the numerator is $\lfloor 3k/2 \rfloor$. This is because the smallest possible permutation size for a pop-stacked permutation with k runs can only be obtained by alternating runs of lengths 1 and 2.

It is even possible to (experimentally) observe further structure in these P_k, for example they have the following partial fraction decomposition:

$$P_k(z) = -1 + \frac{1}{1-kz} + \sum_{j=1}^{k-1} \frac{N_{k,j}(z)}{(1-(k-j)z)^{j+1}}, \tag{4}$$

where $N_{k,j}$ is a polynomial of degree j in z and $2j$ in k:

$$\begin{aligned}
N_{k,1}(z) =& 3(k-1) - (k-1)(3k-4)z,\\
N_{k,2}(z) =& -\frac{1}{2}(3k-4)(3k-11) + \left(9k^3 - 57k^2 + 102k - 44\right)z + \left(-\frac{9k^4}{2} + \frac{69k^3}{2} - 90k^2 + 90k - 26\right)z^2,\\
N_{k,3}(z) =& \left(\frac{9k^3}{2} - \frac{81k^2}{2} + 102k - 70\right) + \left(-\frac{27k^4}{2} + 153k^3 - \frac{1149k^2}{2} + 793k - 256\right)z + \left(\frac{27k^5}{2} - \frac{369k^4}{2} + \frac{1845k^3}{2} - \frac{4067k^2}{2} + 1814k - 406\right)z^2\\
&+ \left(-\frac{9k^6}{2} + 72k^5 - 450k^4 + 1379k^3 - \frac{4223k^2}{2} + 1423k - 244\right)z^3,\\
N_{k,4}(z) =& -\frac{27k^4}{8} + \frac{189k^3}{4} - \frac{1737k^2}{8} + \frac{1533k}{4} - 214 + \left(\frac{27k^5}{2} - 234k^4 + \frac{2943k^3}{2} - 4065k^2 + 4536k - 1184\right)z\\
&+ \left(-\frac{81k^6}{4} + \frac{837k^5}{2} - \frac{13383k^4}{4} + \frac{25887k^3}{2} - 24395k^2 + 19081k - 3418\right)z^2\\
&+ \left(\frac{27k^7}{2} - 324k^6 + \frac{6291k^5}{2} - 15777k^4 + 43070k^3 - 61182k^2 + 37856k - 4956\right)z^3\\
&+ \left(-\frac{27k^8}{8} + \frac{369k^7}{4} - \frac{8433k^6}{8} + \frac{26061k^5}{4} - 23493k^4 + 49493k^3 - 57030k^2 + 29420k - 2968\right)z^4.
\end{aligned}$$

We listed the first few values of these polynomial $N_{k,j}$ in case that some clever mind could find the general pattern for any j. We failed to find a generic closed-form formula, but it is noteworthy that the decomposition of Formula (4) has some similarities with the partial fraction decomposition of Eulerian numbers.

Proposition 8. *Consider the Eulerian numbers $\left\langle {n \atop k} \right\rangle$, defined as the number of permutations of size n that have precisely k runs*[iii]*. The generating functions $E_k(z)$ for the columns of the Eulerian number triangle have the following partial fraction decomposition*

$$E_k(z) = \sum_{n \geq 0} \left\langle {n \atop k} \right\rangle z^n = \frac{1}{1-kz} + \sum_{j=1}^{k-1} \frac{(-1)^j((k-j)z)^{j-1}}{(1-(k-j)z)^{j+1}}. \tag{5}$$

Proof: We were surprised to find no trace of this nice formula in the literature, so we now give a short proof of it. First, inserting $n+1$ into a permutation of size n with k runs either enlarges

[iii] Sometimes, Eulerian numbers are defined such that $\left\langle {n \atop k} \right\rangle$ enumerates all permutations of size n that have k descents. Both cases (counting with respect to runs vs. counting with respect to descents) can be obtained from each other by shifting k by one.

one of the runs if $n+1$ is inserted at its end, or it creates a new run in all other cases; this gives the classical recurrence

$$\left\langle {n+1 \atop k} \right\rangle = (n+1-k)\left\langle {n \atop k-1} \right\rangle + k\left\langle {n \atop k} \right\rangle \qquad \text{for } n,k \geq 1, \text{ with } \left\langle {1 \atop 1} \right\rangle = 1. \tag{6}$$

From it, it is easy to get by induction that

$$\left\langle {n \atop k} \right\rangle = \sum_{j=0}^{k-1} (-1)^j (k-j)^n \binom{n+1}{j}.$$

We refer to [17,24] for thorough surveys on Eulerian numbers, and possible alternative proofs.

Then, rewriting the Eulerian numbers in $E_k(z)$ by means of this formula and carrying out some algebraic manipulations yields

$$\begin{aligned} E_k(z) &= \sum_{n\geq 0} \left(\sum_{j=0}^{k-1} (-1)^j (k-j)^n \binom{n+1}{j} \right) z^n \\ &= \frac{1}{1-kz} + \sum_{j=1}^{k-1} \left((-1)^j \sum_{n\geq j-1} (k-j)^n \binom{n+1}{j} z^n \right) \\ &= \frac{1}{1-kz} + \sum_{j=1}^{k-1} \left((-1)^j ((k-j)z)^{j-1} \sum_{n\geq 0} \binom{n+j}{j} ((k-j)z)^n \right) \\ &= \frac{1}{1-kz} + \sum_{j=1}^{k-1} \frac{(-1)^j ((k-j)z)^{j-1}}{(1-(k-j)z)^{j+1}}. \end{aligned}$$

This proves the partial fraction decomposition. □

The partial fraction decomposition (5) allows us to make statements about the structure of $E_k(z)$. We now list several of these properties, comparing them with our observations for the generating functions $P_k(z)$.

- The degree of the numerator of $E_k(z)$ is $\binom{k+1}{2} - 1$, while for $P_k(z)$ we conjecture it to be $\binom{k+1}{2}$.
- The leading coefficient in the numerator of $E_k(z)$ is $\pm\frac{1}{k}\prod_{m=1}^{k} m!$ (only the first summand in (5) contributes towards the maximum degree), and for $P_k(z)$ we conjecture it to be $\pm\prod_{m=1}^{k} m!$.
- For $E_k(z)$, summing the coefficients in the numerator yields $\pm\prod_{m=1}^{k-1} m!$ (setting $z=1$ in the numerator of $E_k(z)$ only collects contributions from the last summand in (5)), for $P_k(z)$ we conjecture that it is $\pm 2\prod_{m=1}^{k-1} m!$.

While we think that it is unlikely that the sequence $p_{n,k}$ satisfies a simple linear recurrence relation as in (6), we derive in the next section a more complex recurrence scheme that depends on some additional parameters.

2.3 A functional equation for pop-stacked permutations and the corresponding polynomial-time algorithm for the enumeration

As noted above, the counting sequence $(p_n)_{n\geq 0}$ is hard to compute directly without introducing additional parameters. In [4], we mentioned that a generating tree approach leads to an efficient enumeration algorithm for pop-stacked permutations; it relies on an additional parameter which is either the number of runs, or the final value of the permutation. Such generating tree approaches lead to polynomial time algorithms for computing p_n (see [8] for further examples of enumerations via generating tree approaches).

When the additional parameter is the number of runs, this gives a recurrence which encodes the addition of a new run of length k to the end of a given permutation of size n (and relabels this concatenation to get a permutation of size $n+k$). The cost of this approach is analysed in [14], and was implemented with care, allowing the computation of number of pop-stacked permutations of size n, for $n \geq 1000$ with a computer cluster.

When the additional parameter is the final value of the permutation, this gives an approach than we now detail in this section. Based on a generation strategy where either one or two elements are added to a given permutation, we consider the corresponding generating tree. With this strategy, we can derive an appropriate recurrence which also allows us to compute the sequence in polynomial time, and we can also obtain a functional equation for the corresponding multivariate generating function.

In order for our generating approach to work, we need to keep track of a few additional parameters in pop-stacked permutations. Let n, k, a, b, c be non-negative integers such that $k \leq n$ and $1 \leq a \leq b \leq c \leq n$. Let the set $\mathcal{P}_{n,k;a,b,c}$ denote all pop-stacked permutations where

- n denotes the length of the permutation,
- k denotes the number of runs,
- a and c denote the smallest and largest element of the last run, respectively,
- and if $a < c$, then b is the second-largest element of the last run – else, if $a = c$ then also $b := a = c$.

Just to give some examples, the permutation 26|1345 is contained in $\mathcal{P}_{6,2;1,4,5}$, 14|25|36 is contained in $\mathcal{P}_{6,3;3,3,6}$, and $12356|4 \in \mathcal{P}_{6,2;4,4,4}$.

The following set of rules describes how, starting from two initial permutation sets $\mathcal{P}_{1,1;1,1,1} = \{1\}$ and $\mathcal{P}_{2,1;1,1,2} = \{12\}$, all other sets $\mathcal{P}_{n,k;a,b,c}$ can be generated. This heavily relies on the characterization of pop-stacked permutations as given in Theorem 3.

(Rule 1) A new run consisting of a single element is added to the end of all generated permutations. For all integers i with $a+1 \leq i \leq c$, this operation corresponds to an injection $\mathcal{P}_{n,k;a,b,c} \hookrightarrow \mathcal{P}_{n+1,k+1;i,i,i}$.

(Rule 2) A new run consisting of two elements is added to the end of all generated permutations. For all integers $i < j$ with $1 \leq i \leq c$ and $a+2 \leq j \leq n+2$, this corresponds to an injection $\mathcal{P}_{n,k;a,b,c} \hookrightarrow \mathcal{P}_{n+2,k+1;i,i,j}$.

(Rule 3) Insert a new second-largest element into the last run of all generated permutations. For all integers i with $b+1 \leq i \leq c$, this corresponds to the injection $\mathcal{P}_{n,k;a,b,c} \hookrightarrow \mathcal{P}_{n+1,k;a,i,c+1}$.

Proposition 9. *The algorithm described above, starting with $\mathcal{P}_{1,1;1,1,1}$ and $\mathcal{P}_{2,1;1,1,2}$ and iteratively applying Rules 1, 2, and 3 generates all pop-stacked permutations in a unique way.*

Proof: In order to see that all permutations are generated by this strategy we study the corresponding inverse operation. Given a pop-stacked permutation σ of length n that is different from 1 or 12, we consider its last run. Then we carry out the following operations, based on the length of the final run:

- In case the last run is of length 1 or 2 it is simply deleted. This reverses Rules 1 and 2.
- Otherwise, if the run is of length at least 3, the second-largest (and therefore penultimate) element is deleted. This is the reversal of Rule 3.

After proper relabelling, this results in a shorter permutation $\tilde{\sigma}$ that is still pop-stacked. Applying the appropriate expansion rule to $\tilde{\sigma}$ constructs a set of "successors" also containing σ. Observe that by the nature of the expansion rules, only pop-stacked permutations can be generated (as long as we start with a pop-stacked permutation) as it is made sure that the last and the penultimate run always overlap.

This proves that for every permutation (apart from 1 and 12) a unique predecessor in the generating tree can be found. The permutations 1 and 12 are special in the sense that they are the only permutations for which the above strategy does not yield a well-defined result. At the same time, this implies that when starting at any other permutation σ and iterating the procedure of finding the predecessor eventually leads to either 1 or 12. This proves that when starting with these permutations (in our notation, this corresponds to the sets $\mathcal{P}_{1,1;1,1,1}$ and $\mathcal{P}_{2,1;1,1,2}$), all other pop-stacked permutations can be generated by applying Rules 1, 2, and 3 iteratively. □

Now, let us turn our focus from generating these permutations to their enumeration. Let $p_{n,k;a,b,c} := |\mathcal{P}_{n,k;a,b,c}|$, the number of pop-stacked permutations in $\mathcal{P}_{n,k;a,b,c}$. With the notation from Section 2.2, we have

$$p_{n,k} = \sum_{a,b,c\geq 0} p_{n,k;a,b,c} \qquad \text{and} \qquad p_n = \sum_{k\geq 0} p_{n,k}.$$

The generating tree approach can be utilized to derive a functional equation for the associated multivariate generating function

$$P(z,u,v_1,v_2,v_3) = \sum_{\substack{n,k\geq 0\\ a,b,c\geq 0}} p_{n,k;a,b,c} z^n u^k v_1^a v_2^b v_3^c, \tag{7}$$

as well as a recurrence scheme the numbers $p_{n,k;a,b,c}$.

We begin with the functional equation for $P(z,u,v_1,v_2,v_3)$, which is proved by translating the expansion rules from above to the level of algebraic operations on generating functions.

Theorem 10. *The multivariate generating function $P(z,u,v_1,v_2,v_3)$ as given in (7) satisfies the functional equation*

$$\begin{aligned} P(z,u,v_1,v_2,v_3) = 1 &+ zuv_1v_2v_3 + z^2uv_1v_2v_3^2 \\ &+ \frac{zuv_1v_2v_3}{1-v_1v_2v_3}\left(1-\frac{zv_1v_2v_3}{1-v_1v_2}\right)\big(P(z,u,v_1v_2v_3,1,1) - P(z,u,1,1,v_1v_2v_3)\big) \\ &+ \frac{z^2uv_1v_2v_3^2}{(1-v_1v_2)(1-v_3)}\big(P(z,u,v_3,1,1) - P(z,u,1,1,v_1v_2v_3)\big) \\ &+ \frac{z^2uv_1v_2v_3^3}{(1-v_1v_2)(1-v_3)}\big(P(zv_3,u,1,1,v_1v_2) - P(zv_3,u,1,1,1)\big) \\ &+ \frac{zv_2v_3}{1-v_2}\big(P(z,u,v_1,v_2,v_3) - P(z,u,v_1,1,v_2v_3)\big). \end{aligned} \tag{8}$$

Proof: We begin by observing that the first three summands on the right-hand side of (8), $1 + zuv_1v_2v_3 + z^2uv_1v_2v_3^2$, correspond to the empty permutation, as well as the permutations 1 and 12. These are precisely the ones that cannot be generated by expanding another permutation, so they have to be added separately.

For the remaining summands we observe that on the level of generating functions, each of the expansion rules acts as a linear operator on the coefficients. Thus, we consider the effect of each of the expansion rules to the monomial $z^nu^kv_1^av_2^bv_3^c$, and see how this translates back in terms of the generating function. While the corresponding calculations are not too difficult, they can be a bit tedious—which is why we choose to illustrate it for one of the rules, and give only the results for the remaining ones.

Let us first consider Rule 3, which describes the injection $\mathcal{P}_{n,k;a,b,c} \hookrightarrow \mathcal{P}_{n+1,k;a,i,c+1}$ for $b+1 \le i \le c$. Starting from a given permutation whose associated monomial is $z^nu^kv_1^av_2^bv_3^c$, this rule generates longer permutations by inserting a new second-largest element into the last run. On the level of monomials, this corresponds to the map

$$z^nu^kv_1^av_2^bv_3^c \mapsto \sum_{i=b+1}^{c} z^{n+1}v_1^av_2^iv_3^{c+1} = z^{n+1}v_1^av_2\frac{v_2^b - v_2^c}{1-v_2}v_3^{c+1}.$$

Exploiting the linearity of this operator, and rewriting it terms of the generating function yields a contribution of

$$\frac{zv_2v_3}{1-v_2}\big(P(z,u,v_1,v_2,v_3) - P(z,u,v_1,1,v_2v_3)\big)$$

from Rule 3. Similarly, Rule 1 contributes the summand

$$\frac{zuv_1v_2v_3}{1-v_1v_2v_3}\big(P(z,u,v_1v_2v_3,1,1) - P(z,u,1,1,v_1v_2v_3)\big), \tag{9}$$

and Rule 2 corresponds to the remaining terms on the right-hand side of (8) that have not been mentioned yet.

Summing the initial values as well as the contributions from each of the expansion rules then proves the functional equation. □

This allows us to obtain a recurrence relation for $p_{n,k;a,b,c}$ as given in the following theorem. Therein, for the sake of simplicity, we make use of the Iverson bracket [expr], a notation popularized in [20], which evaluates to 1 if expr is a true expression, and 0 otherwise.

Theorem 11 (Polynomial-time computations for pop-stacked permutations)**.** *Let N, K, A, B, C be non-negative integers. The number of pop-stacked permutations of length N with K runs where the last run starts with A, ends with C, and where B is either the penultimate element (if it exists) or coincides with A and C as given by $p_{N,K;A,B,C}$ satisfies the recurrence relation*

$$\begin{aligned} p_{N,K;A,B,C} &= [A = B = C] \sum_{a=1}^{A-1} \sum_{b=1}^{N-1} \sum_{c=A}^{N-1} p_{N-1,K-1;a,b,c} \\ &\quad + [A = B < C] \sum_{a=1}^{C-2} \sum_{b=1}^{N-2} \sum_{c=A}^{N-2} p_{N-2,K-1;a,b,c} \\ &\quad + [A < B < C] \sum_{b=A}^{B-1} p_{N-1,K;A,b,C-1} \end{aligned} \tag{10}$$

together with the initial conditions $p_{1,1;1,1,1} = p_{2,1;1,1,2} = 1$.

The number of pop-stacked permutations of length n with k runs is thus $p_{n,k} = \sum_{a=1}^{n} \sum_{b=a}^{n} \sum_{c=b}^{n} p_{n,k;a,b,c}$, and the number of pop-stacked permutations of length n is thus $p_n = \sum_{k=1}^{n} p_{n,k}$, which can be computed with $\sim n^4/8$ time-complexity and $\sim n^3/3$ memory-complexity.

Proof: We can either go back to the combinatorial description of the generation of permutations via the generating tree approach, or consider the functional equation (8) and extract the coefficient of the monomial $z^N u^K v_1^A v_2^B v_3^C$ on both sides in order to obtain the recurrence relation (10) for $p_{n,k;a,b,c}$. Now, for the complexity analysis, observe that the triple sums appearing in the first and second branch of (10) can be computed more efficiently by defining the auxiliary sequence

$$d_{N,K,a,A} = \sum_{b=1}^{N} \sum_{c=A}^{N} p_{N,K;a,b,c},$$

and note that this sequence can be computed with a backward recurrence A from $N-1$ to 1 via

$$d_{N,K,a,A} = d_{N,K,a,A+1} + \sum_{b=a}^{A} p_{N,K;a,b,A}. \tag{11}$$

Furthermore, observe that we can actually rewrite the third branch of the recurrence (10) (for $A < B-1$ and $B < C$) as

$$\sum_{b=A}^{B-1} p_{N-1,K;A,b,C-1} = p_{N,K;A,B-1,C} + p_{N-1,K;A,B-1,C}.$$

We are now able to rewrite the recurrence in the form

$$\begin{aligned} p_{N,K;A,B,C} = [A = B = C] \sum_{a=1}^{A-1} d_{N-1,K-1,a,A} \\ + [A = B < C]\Big(p_{N-1,K;A,A,A} + \sum_{a=A}^{C-2} d_{N-2,K-1,a,A}\Big) \\ + [A < B < C]\Big([A < B-1](p_{N,K;A,B-1,C} + p_{N-1,K;A,B-1,C}) \\ + [A = B-1]p_{N-1,K;A,A,C-1}\Big). \end{aligned} \tag{12}$$

This approach allows computing $p_{n,k}$ for $1 \le k \le n \le N$ in $O(N^5)$ arithmetic operations. Furthermore, observe that the number of runs is actually not relevant in the recurrence. If we were only interested in p_n, the number of pop-stacked permutations of length n, then we could drop this additional parameter: this allows computing p_n for $1 \le n \le N$ in $O(N^4)$ arithmetic operations, with $O(N^3)$ simultaneous allocations in memory. Actually, we can be even more precise and obtain the asymptotic main term of the number of operations: In the computation of p_n for $1 \le n \le N$, the branches of the recurrence (12) are visited for all $1 \le A \le B \le C \le N$. Investigating these branches more closely reveals that while the case of $A = B = C$ is asymptotically negligible for the main term, both the case of $A = B < C$ as well as $A < B < C$ contribute $\sim N^4/24$ to the total number of additions required. Finally, adding all $p_{n;A,B,C}$ for $1 \le A \le B \le C$ and $1 \le n \le N$ contributes another $\sim N^4/24$ additions—which means that in total the number of additions behaves like $N^4/8$. Similar considerations show that this approach requires $\sim N^3/3$ simultaneous memory allocations. □

A straightforward implementation of the optimized recurrence (12) in SageMath [29] that stores relevant intermediate results in cache memory makes it possible to compute the first 100 terms of the sequence p_n on a standard desktop PC in less than 3 minutes. Figure 7 illustrates the number of additions carried out by our strategy and confirms our assertion regarding the asymptotic main term.

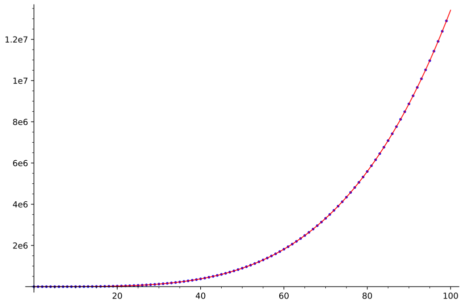


Fig. 7: Number of additions required to compute p_n for $1 \le n \le 100$. The experimental cost (blue dots) nicely fits a quartic polynomial (in red) whose main term is $n^4/8$.

Unfortunately however, neither the recurrence scheme nor the functional equation for the generating function yield sufficient leverage to carry out an exact analysis of the asymptotic behaviour of the sequence p_n. In the following section, we provide a lower bound for the growth of p_n and give pointers to experimental observations.

2.4 Asymptotics of pop-stacked permutations

In many cases, the growth of restricted families of permutations is much less than $n!$, and is just exponential: for example, by the Stanley–Wilf conjecture (proved by Marcus and Tardos [23]), this always holds for permutation classes defined by *classical* forbidden patterns. It is natural to ask whether this is the case for pop-stacked permutations. We now prove that they in fact grow much faster.

Theorem 12 (Superexponential growth of pop-stacked permutations)**.** *The asymptotic growth of the number of pop-stacked permutations is at least* $\exp(n\ln(n) - n\ln(2) - n + o(n))$.

Proof: We achieve this by constructing an explicit class of permutations, as follows. Assume that n is even, and consider any pair of permutations, π and τ, each of size $n/2$. *Intertwine* them as shown in Figure 8, i.e., consider the permutation σ of size n defined by

$$\text{for } 1 \le j \le n/2: \quad \sigma(2j-1) = \pi(j),\ \sigma(2j) = \tau(j). \tag{13}$$

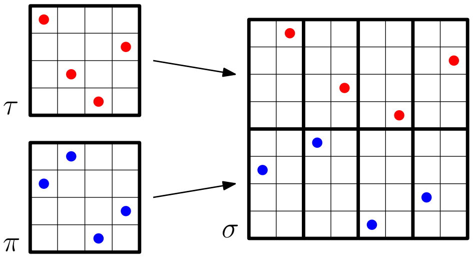


Fig. 8: Intertwining two permutations gives a pop-stacked permutation and proves Theorem 12.

It is easy to see that such σ is necessarily pop-stacked (it clearly satisfies the overlapping run condition from Theorem 3). Since the mapping $(\pi,\tau) \to \sigma$ is injective, we conclude that we have at least $((n/2)!)^2$ pop-stacked permutations of size n. Then, Stirling's approximation for the factorial leads to the theorem. Note that the fact that the above argument holds for even n only is not a restriction: by inserting $n+1$ at the end of any pop-stacked permutation of length n, one obtains a new pop-stacked permutation of size $n+1$; therefore the sequence p_n is strictly increasing, and thus the Stirling bound given in the claim remains valid for odd n. □

Due to the nature of the counting sequence and the fact that we have no appropriate representation of the associated generating function, an exact asymptotic analysis of the growth behaviour of the sequence remains a challenge. The above theorem gives a lower bound of $.5^n n!$, it is also possible to compare with André's alternating permutations to get a lower bound of $.63^n n!$ (see e.g. [16, p. 5]). Note that the authors of [14] carried out an experimental analysis using automated fitting and differential approximation. Their analysis of the counting sequence led them to conjecture that the corresponding exponential generating function possesses an infinite number of singularities, thus implying non-D-finiteness. Ultimately, they conjecture that the asymptotic growth of the sequence p_n counting pop-stacked permutations of size n is

$$p_n \sim (0.695688\ldots) \cdot (0.898118\ldots)^n \cdot n!;$$

see [14, Section 3.3] for more details.

3 Best cases of flip-sort: permutations with low cost

3.1 Notation and earlier work

Some of the earlier work on pop-stack sorting dealt with permutation with **low** cost. The following notation was introduced there.

Definition 13. A permutation π is *k-pop-stack-sortable* if $\mathsf{cost}(\pi) \leq k$ (or, equivalently, if $\mathrm{Im}(T^k)(\pi) = \mathsf{id}$).

1- and 2-pop-stack-sortable permutations are shown by the red and the blue frames in the pop-stack-sorting tree in Figure 2.

Several results concerning k-pop-stack-sortability are known:

- Avis and Newborn [7] proved that a permutation is 1-pop-stack-sortable if and only if it is layered[(iv)].
- Pudwell and Smith [26] found a structural characterization of 2-pop-stack-sortable permutations, as well as a bijection between such permutations and *polyominoes on a twisted cylinder of width* 3 (see [2]). Moreover, they proved that the generating function for 2-pop-stack-sortable permutations is rational (we shall reprove their result below).
- Claesson and Guðmundsson [13] generalized the latter result showing that for each fixed k, the generating function for k-pop-stack-sortable permutations is rational.

3.2 The pre-image of 1-pop-stack-sortable permutations

We begin with a nice enumerative result in which the notion of 1- and 2-pop-stack-sortable permutations is combined with our notion of pop-stacked permutations. Specifically, it concerns those 1-pop-stack-sortable permutations that belong to the image of T (thus, corresponding to internal nodes in the right column before the identity in the pop-stack tree from Figure 2).

Theorem 14. *1. Let $\mathcal{LI}_n$ be the set of pop-stacked layered permutations (or, equivalently, of 1-pop-stack-sortable permutations that belong to* $\mathrm{Im}(T)$*) of size n. The generating function for $(|\mathcal{LI}_n|)_{n\geq 1}$ is*

$$x + \frac{(x+x^2)^2}{1-(x+x^2+x^3)} = x + x^2 + 3x^3 + 5x^4 + 9x^5 + 17x^6 + 31x^7 + \ldots. \qquad (14)$$

Consequently, this family is enumerated by "tribonacci numbers" (`A000213`).

2. *Let $\tau \in \mathcal{LI}_n$. All the permutations π such that $T(\pi) = \tau$ can be constructed by the following procedure:*

   - *Put* primary bars *at all descents of τ.*
   - *Optionally, put* secondary bars *at some descents so that each primary bar has at least one neighbouring position without any bar.*
   - *Reverse all the blocks determined by the bars.*

[(iv)] A layered permutation is an increasing sequence of falls.

Proof:

1. Let τ be a layered permutation of size n. Partition it into falls. Refer to the first and the last falls as *outer falls*, to all other falls as *inner falls*. For $n \geq 3$ it is easy to see that we have non-overlapping adjacent runs if and only if there is an outer fall of size ≥ 3 or an inner fall of size ≥ 4. For $n = 2$ the permutation 12 is pop-stacked, and the permutation 21 is not. It follows that for $n \geq 2$ we have $\tau \in \mathcal{LI}_n$ if and only if it has at least two falls, the outer falls being of size 1 or 2, the inner falls of size 1 or 2 or 3. This implies the generating function (14).
2. As τ is obtained from π by reversing all its falls, π is obtained from τ by reversing its runs – possible, further partitioned. That is, we must first partition τ into runs (by primary bars), then optionally further partition the runs (by secondary bars), then reverse all the blocks.

   We need to prove the condition on secondary bars. Suppose a primary bar separates a descent $b_i|b_{i+1}$ (we have $b_i > b_{i+1}$). If we put bars at both adjacent gaps, $|b_i|b_{i+1}|$, then we have the same values in positions i and $i+1$ in π. However, this would be a descent, that is, a part of a fall in π, and it should be reversed when we apply T. Otherwise, both sub-runs (that from the left and that from the right of the primary bar) will yield, upon reversal, two distinct falls in π, that will recover these sub-runs when we apply T. □

Example. Let $\tau = 13254687$. The primary bars are inserted at the descents: 13|25|468|7. All the possibilities to insert secondary bars (for the sake of clarity, we indicate them by ·) are 13|25|468|7, 1·3|25|468|7, 13|2·5|468|7, 13|25|4·68|7, 1·3|25|4·68|7, which gives all the pre-images of τ: 31|52|864|7, 1·3|52|864|7, 31|2·5|864|7, 31|52|4·86|7, 1·3|52|4·86|7. Further, it can be shown that the number of pre-images of $\tau \in \mathcal{LI}_n$ is always the product of some Fibonacci numbers depending on the sequence of sizes of falls of τ.

3.3 2-pop-stack-sortable permutations and lattice paths

In this section we extend some of the results by Pudwell and Smith [26]. Namely, we reprove one of their theorems and prove (a generalization of) one of their conjectures. All this is done via the uniform framework of lattice paths. By doing this, we not only construct a bijection between 2-pop-stack sortable permutations and lattice paths, but also benefit form the fact that the theory of lattice paths is well developed (see e.g. [9]), thus offering additional structural insight.

First, we reprove the theorem by Pudwell and Smith in a more combinatorial way. Then we construct a bijection between 2-pop-stack-sortable permutations and a certain family of lattice paths, which enables us to prove two of their conjectures.

Theorem 15. *For $n \geq 1$, $0 \leq k \leq n-1$, let $a_{n,k}$ be the number of 2-pop-stack-sortable permutations of size n with exactly k ascents.*

1. ***(Pudwell and Smith [26], Thm. 2.5)*** *Let $A(x,y)$ be the bivariate generating function $A(x,y) = \sum a_{n,k}x^n y^k$. Then we have*

$$A(x,y) = \frac{x(1+x^2y)}{1-x-xy-x^2y-2x^3y^2}. \tag{15}$$

2. ***(For*** $k = 0$***, Pudwell and Smith [26], Conj. 4.3)*** *Let* $k \geq 0$*. The generating functions for "diagonal-parallel" arrays are*

$$\begin{aligned} D_k(x) &= \sum_{n\geq 0} a_{2n+k+1,n+k}x^n = \sqrt{\frac{1+x}{1-7x}}\left(\frac{1-x-\sqrt{(1+x)(1-7x)}}{2x}\right)^k, \\ D_{-k}(x) &= \sum_{n\geq 0} a_{2n+k+1,n}x^n = \sqrt{\frac{1+x}{1-7x}}\left(\frac{1-x-\sqrt{(1+x)(1-7x)}}{2x(1+2x)}\right)^k. \end{aligned} \tag{16}$$

For the "diagonal array" $a_{2n+1,n}$ *(that is, for* $k = 0$*) we have explicit formulas*

$$a_{2n+1,n} = \sum_{i=0}^{n-1}(-1)^i 2^{n-i}\binom{2(n-i)}{n-i}\binom{n-1}{i} = \sum_{k\geq 0}\binom{n}{2k}\binom{2k}{k}2^{2k+1}3^{n-2k-1}\left(2-\frac{k}{n}\right). \tag{17}$$

The two-dimensional array $a_{n,k}$ is shown in Figure 9. The nth row ($n \geq 1$) contains coefficients $a_{n,0}, a_{n,1}, \ldots, a_{n,n-1}$. The "diagonal-parallel" arrays from part 2 are vertical in this drawing ($k = 0$ corresponds to $1, 4, 20, 116, \ldots$, $k = 1$ to $1, 8, 48, 296, \ldots$, etc.).

$$\begin{array}{ccccccccccccccccc}
& & & & & & & & 1 & & & & & & & & \\
& & & & & & & 1 & & 1 & & & & & & & \\
& & & & & & 1 & & 4 & & 1 & & & & & & \\
& & & & & 1 & & 6 & & 8 & & 1 & & & & & \\
& & & & 1 & & 8 & & 20 & & 12 & & 1 & & & & \\
& & & 1 & & 10 & & 36 & & 48 & & 16 & & 1 & & & \\
& & 1 & & 12 & & 56 & & 116 & & 92 & & 20 & & 1 & & \\
& 1 & & 14 & & 80 & & 224 & & 296 & & 152 & & 24 & & 1 & \\
\bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \bullet
\end{array}$$

Fig. 9: The two-dimensional array $a_{n,k}$, $n \geq 1$, $0 \leq k \leq n-1$. The coefficient $a_{n,k}$ is the number of 2-pop-stack-sortable permutations with exactly k ascents.

Proof:

1. It is shown in [26, Lemma 2.1] that a permutation π is 2-pop-stack-sortable if and only if for each pair of its adjacent falls, F_i and F_{i+1}, one has $\max(F_i) \leq \min(F_{i+1}) + 1$.[v] Consider an ascent that separates two adjacent falls F_i and F_{i+1}. We say that this ascent is *regular* if $\max(F_i) < \min(F_{i+1})$, and *twisted* if $\max(F_i) = \min(F_{i+1}) + 1$. An ascent can be twisted only when if at least one of these falls, F_i and F_{i+1}, is of size > 1 (equivalently, when at least one of the adjacent gaps is a descent).

   As Pudwell and Smith show, 2-pop-stack-sortable permutations are bijectively encoded by sequences of their ascents and descents (we use a for ascents, d for descents), where for each a that has an adjacent d it is indicated whether this a is regular or twisted. We find it

[v] Notice that layered permutations are similarly characterized by $\max(F_i) = \min(F_{i+1}) - 1$.

convenient to use instead lattice paths: we replace each a by the *up-step* $\mathsf{U} = (1,1)$, d by the *down-step* $\mathsf{U} = (1,-1)$, and *corner-adjacent* U-steps (that is, U-steps that have at least one adjacent D-step) will be bicoloured: U-steps that correspond to a regular ascent will be coloured black, U-steps that correspond to a twisted ascent will be coloured red. In this way we obtain a certain family of Dyck walks. The length of a walk is smaller by 1 than the size of the corresponding permutation, and the final altitude is $\#(\mathsf{a}) - \#(\mathsf{d})$. See Figure 10 for illustration.

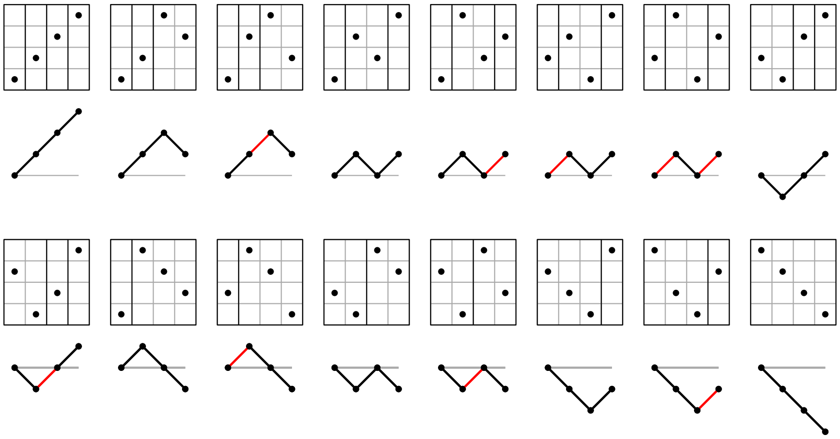


Fig. 10: Encoding of 2-pop-stack-sortable permutations by Dyck walks with bicoloured corner-adjacent U-steps.

To enumerate such walks, we make use of the symbolic method (see [16, Chapter I]) to find a combinatorial specification for (non-coloured) Dyck walks in which corner-adjacent U-steps are marked by y:

$$\mathcal{W} = \epsilon \cup \mathsf{Seq}^{\geq 1}(\mathsf{U}) \times \left(\epsilon \cup y\mathsf{Seq}^{\geq 1}(\mathsf{D}) \times \left(\epsilon \cup \mathsf{U}y \times \mathcal{W}\right)\right) \cup \mathsf{Seq}^{\geq 1}(\mathsf{D}) \times \left(\epsilon \cup \mathsf{U}y \times \mathcal{W}\right).$$

This leads to the following functional equation for the trivariate generating function $W(t,u,y)$, where t is the variable for length, u for final altitude, y for occurrences of corner-adjacent U-steps:

$$W = 1 + \frac{tu}{1-tu}\left(1 + \frac{t/u}{1-t/u}y\left(1+tuyW\right)\right) + \frac{t/u}{1-t/u}\left(1+tuyW\right). \tag{18}$$

This yields the generating function

$$W(t,u,y) = \frac{u(1-t^2(1-y))}{u - t - tu^2 + t^2u(1-y) + t^3u^2y(1-y)}, \tag{19}$$

and the bivariate generating function for those walks in which corner-adjacent U-steps are *bicoloured*, is

$$W(t,u,2) = \frac{u\big(1+t^2\big)}{u-t-tu^2-t^2u-2t^3u^2}. \tag{20}$$

Upon performing the transformation that corresponds to the way in which we rearranged the array of the coefficients, we obtain $xW(x\sqrt{y},\sqrt{y},2) = A(x,y)$. This completes the proof of part 1.

2. First, we find the generating function $E(t,y)$ for excursions — that is, those walks that stay (weakly) above and terminate at the t-axis — in our model of Dyck walks with marked corner-adjacent U-steps. The functional equation for $E(t,y)$ is

$$E = 1 + \frac{t^2\big(y+(E-1)\big)}{1-t^2yE}, \tag{21}$$

which yields

$$E(t,y) = \frac{1-t^2(1-y)-\sqrt{\big(1-t^2(1-y)\big)\big(1-t^2(1+3y)\big)}}{2t^2y}. \tag{22}$$

Therefore, the generating function for Dyck excursions with *bicoloured* corner-adjacent U-steps is

$$E(t,2) = \frac{1+t^2-\sqrt{(1+t^2)(1-7t^2)}}{4t^2}. \tag{23}$$

(In fact, we have $E(t,y) = 1 + \frac{t^2y}{1-t^2}M\Big(\frac{t^2y}{1-t^2}\Big)$, where $M(x)$ is the generating function for Motzkin numbers. This can be shown bijectively, starting from the fact that the number of Dyck excursions of semilength n with exactly k corner-adjacent U-steps, is $\binom{n-1}{k-1}M_{k-1}$, where M_{k-1} is the $(k-1)$st Motzkin number. A similar result on Dyck paths with marked pattern UDU was proved by Sun in [28, Thm. 2.2].)

The "diagonal coefficients" $a_{2n+1,n} = [x^{2n+1}y^n]$ of $A(x,y)$ correspond to coefficients $[t^{2n}u^0]$ of $W(t,u,2)$. Therefore we look for the generating function $B(t,y)$ for *bridges* — that is, those walks that terminate at the t-axis — in our model. It can be routinely found by residue analysis (extracting $[u^0]$ in $W(t,u,y)$). However, we give a more structural proof. We decompose a bridge into an alternating sequence of excursions and anti-excursions (= rotated by 180° excursions that stay weakly below the x-axis). Since we use only U and D steps, such a decomposition is unambiguous. Moreover, the generating function for anti-excursions is the same as for excursions. When we join excursions and anti-excursions, new corner-adjacent U-steps are never created. Therefore we have

$$B(t,y) = \frac{E(t,y)}{1-(E(t,y)-1)} = \frac{\sqrt{1-t^2(1-y)}}{\sqrt{1-t^2(1+3y)}}, \tag{24}$$

and finally

$$B(t,2)=\frac{\sqrt{1+t^2}}{\sqrt{1-7t^2}}=1+4t^2+20t^4+116t^6+708t^8+4452t^{10}+\dots. \tag{25}$$

Notice that

$$B(t,2)=\sqrt{\frac{1+t^2}{1-7t^2}}=\frac{1}{\sqrt{1-\frac{8t^2}{1+t^2}}}=\left.\frac{1}{\sqrt{1-4x}}\right|_{x=2t^2/(1+t^2)},$$

and the coefficients of $1/\sqrt{1-4x}$ are well known to be central binomial coefficients. This directly leads to the first closed-form formula of Equation (17) for $a_{2n+1,n}$. The second equivalent closed-form formula is proved easily via closure properties from holonomy theory; see [25].

Finally, we consider generating functions $W_k(t,2)$ for walks that terminate at fixed altitude k. For unmarked paths we have the classical decomposition $W_k(t)=B(t)(tE(t))^k$. Taking care of corner-adjacent U-steps, we obtain, for $k\geq 0$,

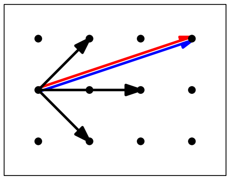
$$W_{+k}(t,2)=\frac{\sqrt{1+t^2}}{\sqrt{1-7t^2}}\left(\frac{1-t^2-\sqrt{(1+t^2)(1-7t^2)}}{2t}\right)^k,$$

and

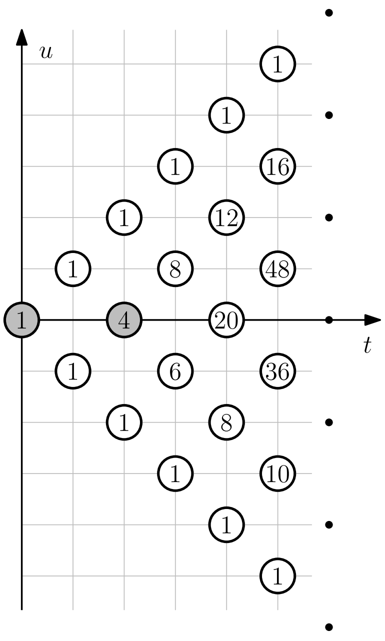
$$W_{-k}(t,2)=\frac{\sqrt{1+t^2}}{\sqrt{1-7t^2}}\left(\frac{1-t^2-\sqrt{(1+t^2)(1-7t^2)}}{2t(1+2t^2)}\right)^k,$$

which, upon due modification, yields (16). □

Finally, if we look again at (20) and rewrite it in the form $W(t,u,2)=\frac{1+t^2}{1-\frac{t}{u}-tu-t^2-2t^3u}$ then it is immediate from the theory of lattice paths that it is also the bivariate generating function for lattice paths that start either at $(0,0)$ or at $(2,0)$, and use steps $(1,-1)$, $(1,1)$, $(2,0)$, and bicoloured $(3,1)$.



These paths can be also used to obtain formulas (16) for diagonal-parallel arrays, and the computations are even easier because being some steps bicoloured does not depend on adjacent steps.



4 Worst-cases of flip-sort: permutations with high cost

> *"Conventional wisdom might say that extremal problems tend to be very difficult to solve exactly if the extreme configurations are so diverse and irregular."*
> Martin Aigner & Günter Ziegler, in *Proofs from THE BOOK*.

This claim by Aigner and Ziegler is concerning an extremal problem in geometry which is in fact solved by understanding the worst cases of flip-sort! We refer to [1, Chapter 12] and [30] for more details on this geometric problem. In this section, we strongly extend these studies by proving structural and enumerative results concerning the evolution of π under flip-sort, while concentrating on permutations with worst costs. We start with some definitions, notation, and basic observations.

4.1 Definitions and notation

By id_n we denote the identity permutation of size n, and by $-\mathsf{id}_n$ we denote its reversal: $\mathsf{id}_n = 12\ldots n$, and $-\mathsf{id}_n = n\ldots 21$.

Definition 16 (layered and skew-layered permutations)**.** The *layered permutation* $\oplus(m_1, m_2, \ldots, m_s)$ is the direct sum of $-\mathsf{id}_{m_1}, -\mathsf{id}_{m_2}, \ldots, -\mathsf{id}_{m_s}$. The *skew-layered permutation* $\ominus(m_1, m_2, \ldots, m_s)$ is the skew sum of $\mathsf{id}_{m_1}, \mathsf{id}_{m_2}, \ldots, \mathsf{id}_{m_s}$.[(vi)] For example, $\oplus(2,1,3) = 213654$ and $\ominus(2,1,3) = 564123$.

Definition 17 (k-shadows)**.** Let π be a permutation of size n, and let k be a fixed number, $1 \le k \le n-1$. Let $S := \{1, \ldots, k\}$, $L := \{k+1, \ldots, n\}$ (S for *small*, L for *large*). The *k-shadow* of π is defined to be the $\{\mathsf{S}, \mathsf{L}\}$-word $\mathsf{shadow}_k(\pi)$ obtained from π by replacing all the numbers from S by S, and all the numbers from L by L. When k is fixed, we shall usually omit the subscript and write just $\mathsf{shadow}(\pi)$. *Example:* for $\pi = 6317524$ we have $\mathsf{shadow}_3(\pi) = \mathsf{LSSLLSL}$.

For $1 \le j \le k$, we denote by $s_j(\pi)$ the position of the jth occurrence of S counted **from left** in $\mathsf{shadow}(\pi)$; for $1 \le j \le n-k$, we denote by $\ell_j(\pi)$ the position of the jth occurrence of L counted **from right** in $\mathsf{shadow}(\pi)$. In the example just considered, we have $s_1(\pi) = 2$, $s_2(\pi) = 3$, $s_3(\pi) = 6$; $\ell_1(\pi) = 7$, $\ell_2(\pi) = 5$, $\ell_3(\pi) = 4$, $\ell_4(\pi) = 1$.

Definition 18 (S- and L-paths, k-wiring diagram)**.** Let π and k be as above. For $1 \le j \le k$, the *jth S-path* is the sequence $(s_j(T^m(\pi)))_{m \ge 0}$. For $1 \le j \le n-k$, the *jth L-path* is the sequence $(\ell_j(T^m(\pi)))_{m \ge 0}$. The *$k$-wiring diagram* is a drawing on the rectangular grid with infinitely many rows, labelled (from above) by $0, 1, 2, \ldots$, that correspond to $\pi, T(\pi), T^2(\pi), \ldots$; and n columns labelled by $1, 2, \ldots, n$. In this drawing, we represent the jth S-path ($1 \le j \le k$) by a polygonal red line connecting the points $(m, s_j(T^m(\pi)))_{m \ge 0}$, and we represent the jth L-path ($1 \le j \le n-k$) by a polygonal blue line connecting the points $(m, \ell_j(T^m(\pi)))_{m \ge 0}$. The jth S-path (resp. S-path) of π will be denoted by $\mathsf{S}_j(\pi)$ (resp. by $\mathsf{L}_j(\pi)$). See Figure 11 for an example.

Remark: Since we show below that after the $(n-1)$st row the paths stay in the same position (namely, the $\mathsf{S}_j(\pi)$ in the position j, and the $\mathsf{L}_j(\pi)$-path in the position $n+1-j$), we draw k-wiring diagrams with only n rows, from that corresponding to π to that corresponding to $T^{n-1}(\pi)$.

[(vi)] Consult *Permutation patterns: basic definitions and notation* by David Bevan [10] for the definitions of direct sum and of skew sum.

π	8	6	4	7	5	2	3	1
$T(\pi)$	4	6	8	2	5	7	1	3
$T^2(\pi)$	4	6	2	8	5	1	7	3
$T^3(\pi)$	4	2	6	1	5	8	3	7
$T^4(\pi)$	2	4	1	6	5	3	8	7
$T^5(\pi)$	2	1	4	3	5	6	7	8
$T^6(\pi)$	1	2	3	4	5	6	7	8
$T^7(\pi)$	1	2	3	4	5	6	7	8

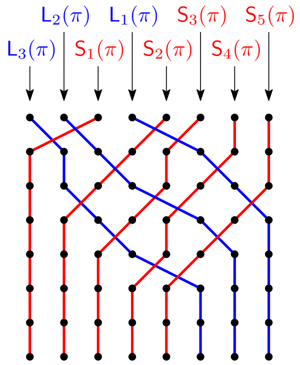


Fig. 11: The k-wiring diagram for $\pi = 86231745$ and $k = 5$.

In the following considerations, the k-wiring diagram of the permutation $\rho_k := \ominus(n-k,k)$ will play an important role. When k is fixed, we shall usually omit the subscript and write just ρ. It is easy to verify that $\mathsf{shadow}(\rho) = \mathsf{L}^{n-k}\mathsf{S}^k$, and that for each $m \geq 0$ the word $\mathsf{shadow}(T^{m+1}(\rho))$ is obtained from $\mathsf{shadow}(T^m(\rho))$ by replacing each consecutive occurrence of LS by SL. This yields

$$\begin{aligned} s_j(T^m(\rho)) &= \begin{cases} n-k+j, & 0 \leq m \leq j-1, \\ n-k+2j-m-1, & j-1 \leq m \leq n-k+j-1, \\ j, & n-k+j-1 \leq m; \end{cases} \\ \ell_j(T^m(\rho)) &= \begin{cases} n-k+1-j, & 0 \leq m \leq j-1, \\ n-k-2j+m+2, & j-1 \leq m \leq k+j-1, \\ n+1-j, & k+j-1 \leq m. \end{cases} \end{aligned} \tag{26}$$

Accordingly, the k-wiring diagram of ρ has a very distinctive shape, with S- and L-paths typically consisting of three segments. This is illustrated in Figure 12 (since we later combine the k-wiring diagram of the fixed permutation ρ with that of an arbitrary permutation π, the S- and L-paths of ρ will be shown by thick pink and light-blue lines). Finally, we notice that, since the sets S and L are already sorted in ρ, all the values along $\mathsf{S}_j(\rho)$ are j, and all the values along $\mathsf{L}_j(\rho)$ are $n+1-j$, so that for $\rho = \ominus(n-k,k)$ the k-wiring diagram coincides with the *usual* wiring diagram where paths connect occurrences of the same value.

Definition 19 (Poset of words)**.** Assume $1 \leq k \leq n-1$. Let $\mathcal{SL}(k, n-k)$ be the set of words of length n with k occurrences of S and $n-k$ occurrences of L. For $\lambda, \mu \in \mathcal{SL}(k, n-k)$ we write $\lambda \preccurlyeq \mu$ if for each j, $1 \leq j \leq k$, the position of the jth S in λ is weakly to the left from the position of the jth S in μ. Equivalently, $\lambda \preccurlyeq \mu$ if for each j, $1 \leq j \leq n-k$, the position of the jth L in λ is weakly to the right from the position of the jth L in μ [(vii)]. For example, $\mathsf{SSSLL} \preccurlyeq \mathsf{SLSLS} \preccurlyeq \mathsf{LSLSS}$. It is easy to verify that $\preccurlyeq$ is a partial order relation, and that these two definitions are indeed equivalent.

[(vii)] We remind that the occurrences of S are counted **from left**, and the occurrences of L are counted **from right**.

ρ	6	7	8	1	2	3	4	5
$T(\rho)$	6	7	1	8	2	3	4	5
$T^2(\rho)$	6	1	7	2	8	3	4	5
$T^3(\rho)$	1	6	2	7	3	8	4	5
$T^4(\rho)$	1	2	6	3	7	4	8	5
$T^5(\rho)$	1	2	3	6	4	7	5	8
$T^6(\rho)$	1	2	3	4	6	5	7	8
$T^7(\rho)$	1	2	3	4	5	6	7	8

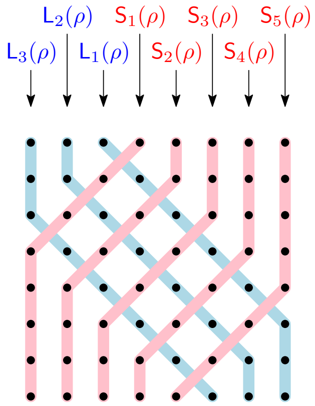


Fig. 12: The k-wiring diagram for $\rho = \ominus(n-k,k)$ for $n=8$, $k=5$.

In fact, $\preccurlyeq$ is the transitive closure of the relation "u is obtained from v by replacing some consecutive occurrence of LS by SL". In particular, it follows that $\mathsf{L}^{n-k}\mathsf{S}^k$ — the maximum element of $\mathcal{SL}(k,n-k)$ — covers just one element: $\mathsf{L}^{n-k-1}\mathsf{SLS}^{k-1}$, and that $\mathsf{S}^k\mathsf{L}^{n-k}$ — the minimum element of $\mathcal{SL}(k,n-k)$ — is covered by just one element: $\mathsf{S}^{k-1}\mathsf{LSL}^{n-k-1}$.

See Figure 13 for the Hasse diagram of $\mathcal{SL}(k,n-k)$ with the partial order $\preccurlyeq$, for $n=6, k=3$ (the words marked by blue colour are the 3-shadows of $\rho, T(\rho), \ldots$).

Definition 20 (bandwidth)**.** The *bandwidth* (also called maximum displacement) of a permutation $\pi = a_1a_2\ldots a_n$ is

$$\mathsf{d}(\pi) := \max_{1\le i\le n} |a_i - i|. \tag{27}$$

The inequality $\mathsf{d}(\pi) \le r$ corresponds to the diagram of π being $(2r+1)$-diagonal. This means that it is entirely contained in the main diagonal and its r shifts to either side. Obviously, for each permutation π of size n, we have $\mathsf{d}(\pi) \le n-1$, and $\mathsf{d}(\pi) = 0$ if and only if $\pi = \mathrm{id}$.

4.2 The worst case: $n-1$ iterations, and a key statistic: bandwidth of permutations

The main result of this section is the following theorem about the bandwidth of a permutation in $\mathrm{Im}(T^m)$. In terms of permutation diagrams, this theorem says that grey areas as in Figure 3 do not contain any points of respective diagrams.

Theorem 21. *If $\tau = b_1b_2\ldots b_n \in \mathrm{Im}(T^m)$ (where $0 \le m \le n-1$), then we have $\mathsf{d}(\tau) \le n-1-m$.*

Notice that for $m = n-1$ we obtain that a permutation in $\mathrm{Im}(T^{n-1})$ has $\mathsf{d} = 0$, and thus we recover Ungar's result mentioned above, $\mathrm{Im}(T^{n-1}) = \{\mathsf{id}\}$. The proof that we give is also a generalization of Ungar's proof of this special case.

The proof of Theorem 21 will follow from the following propositions.

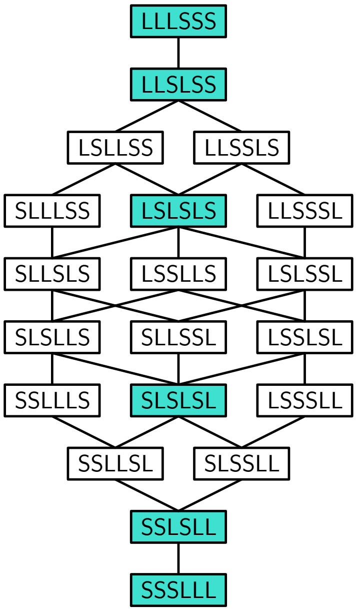


Fig. 13: The Hasse diagram of $\mathcal{SL}(k, n-k)$ with the partial order $\preccurlyeq$, for $n = 6, k = 3$. The k-shadows of $\rho, T(\rho), T^2(\rho), \ldots$ are shown by blue colour.

Proposition 22. *Let σ be a permutation of size n. Let $1 \leq k \leq n-1$. Let $\lambda \in \mathcal{SL}(k, n-k)$ (the poset defined in Definition 13) such that $\lambda \neq \mathsf{S}^k\mathsf{L}^{n-k}$. Let λ' be the word obtained from λ by replacing each consecutive occurrence of* LS *by* SL. *Then we have: If* $\mathsf{shadow}(\sigma) \preccurlyeq \lambda$ *then* $\mathsf{shadow}(T(\sigma)) \preccurlyeq \lambda'$.

Proof: The word $\mathsf{shadow}(T(\sigma))$ is obtained from $\mathsf{shadow}(\sigma)$ by flips of the form $\mathsf{L}^a\mathsf{S}^b \to \mathsf{S}^b\mathsf{L}^a$ (for some $a, b \geq 0$), induced by flips in σ. It follows that for each j ($1 \leq j \leq k$) we have $s_j(\mathsf{shadow}(T(\sigma))) \leq s_j(\mathsf{shadow}(\sigma))$. Moreover, if $s_j(\mathsf{shadow}(\sigma)) = \alpha$ and the $(\alpha - 1)$st position in $\mathsf{shadow}(\sigma))$ is L, then we have $s_j(\mathsf{shadow}(T(\sigma))) < s_j(\mathsf{shadow}(\sigma))$. As for λ: since all the flips in λ are of the form LS → SL, we always have $s_j(\lambda') = s_j(\lambda)$ or $s_j(\lambda') = s_j(\lambda) - 1$.

In terms of k-wiring diagram this means that S-paths are monotone in the sense that, as we scan them from the top to the bottom, they move weakly to the left at each step. Moreover, if there is L before S, then the path that goes through this S will move strongly to the left at the next step.

We now prove the claim. Assume for contradiction that we have $\mathsf{shadow}(\sigma) \preccurlyeq \lambda$ but $\mathsf{shadow}(T(\sigma)) \not\preccurlyeq \lambda'$. Due to the properties just seen, this can only happen if for some j, we have $(\alpha :=)\ s_j(\mathsf{shadow}(\sigma)) = s_j(\mathsf{shadow}(T(\sigma))) = s_j(\lambda) = s_j(\lambda') + 1$ (see Figure 14). Since $s_j(\lambda') = s_j(\lambda) - 1$, the $(\alpha - 1)$st position in λ' is occupied by L. Since $s_j(\mathsf{shadow}(T(\sigma))) = s_j(\mathsf{shadow}(\sigma))$, the $(\alpha - 1)$st position in $\mathsf{shadow}(\sigma)$ is occupied by S. This means $s_{j-1}(\sigma) = \alpha - 1$ and $s_{j-1}(\lambda) < \alpha - 1$, which contradicts $\mathsf{shadow}(\sigma) \preccurlyeq \lambda$. □

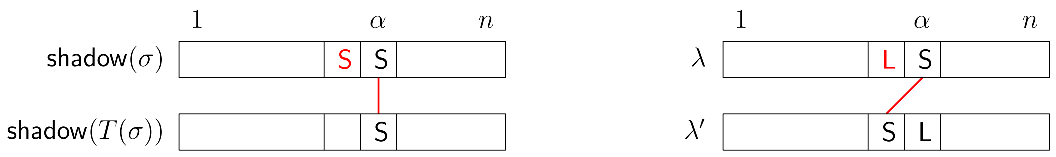


Fig. 14: Illustration to the proof of Proposition 22 (shadow(σ) $\preccurlyeq \lambda \;\Rightarrow\;$ shadow($T(\sigma)$) $\preccurlyeq \lambda$).

The next proposition is a rephrasing of one of the claims in Ungar's proof.

Proposition 23. *Let π be any permutation of size n, and let $1 \leq k \leq n-1$. Let $\rho = \ominus(n-k, k)$ as defined in Section 4.1. Then, for each m ($1 \leq m \leq n-1$), we have for k-shadows* shadow($T^m(\pi)$) $\preccurlyeq$ shadow($T^m(\rho)$).

Proof: First, we have shadow(π) $\preccurlyeq$ shadow(ρ) because in shadow(ρ) the k letters S occupy the last k positions.

Then, the claim follows from Proposition 22 by induction since for $0 \leq m \leq n-2$ we have shadow($T^{m+1}(\rho)$) $=$ (shadow($T^m(\rho)$))$'$. □

In terms of k-wiring diagrams, Proposition 23 says that the paths in the k-wiring diagram of ρ **majorate** the paths in the k-wiring diagram of π in the following sense: for each j ($1 \leq j \leq k$), the path $\mathsf{S}_j(\pi)$ is weakly to the left of the path $\mathsf{S}_j(\rho)$; and similarly, for each j ($1 \leq j \leq n-k$), the path $\mathsf{L}_j(\pi)$ is weakly to the right of the path $\mathsf{L}_j(\rho)$. For illustration, see Figure 15 in which S- and L-paths of π (thin red and blue lines) are shown together with S- and L-paths of ρ (thick pink and light-blue lines).

shadow($T^m(\pi)$), $0 \leq m \leq n-1$									shadow($T^m(\rho)$), $0 \leq m \leq n-1$							
8	6	4	7	5	2	3	1	$\preccurlyeq$	6	7	8	1	2	3	4	5
4	6	8	2	5	7	1	3	$\preccurlyeq$	6	7	1	8	2	3	4	5
4	6	2	8	5	1	7	3	$\preccurlyeq$	6	1	7	2	8	3	4	5
4	2	6	1	5	8	3	7	$\preccurlyeq$	1	6	2	7	3	8	4	5
2	4	1	6	5	3	8	7	$\preccurlyeq$	1	2	6	3	7	4	8	5
2	1	4	3	5	6	7	8	$\preccurlyeq$	1	2	3	6	4	7	5	8
1	2	3	4	5	6	7	8	$\preccurlyeq$	1	2	3	4	6	5	7	8
1	2	3	4	5	6	7	8	$\preccurlyeq$	1	2	3	4	5	6	7	8

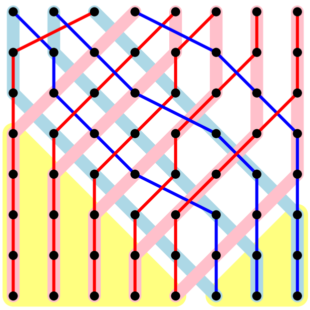


Fig. 15: Illustration to Proposition 23. Left part: For each $m \geq 0$, shadow($T^m(\pi)$) $\preccurlyeq$ shadow($T^m(\rho)$). Right part: The wiring diagrams for π and for ρ shown superimposed. For each j, the red path $\mathsf{S}_j(\pi)$ is weakly to the left of the pink path $\mathsf{S}_j(\rho)$; for each j, the blue path $\mathsf{L}_j(\pi)$ is weakly to the right of the light-blue path $\mathsf{L}_j(\rho)$. The triangular areas (indicated by a yellow colour) only contain S for the left triangle and L for the right triangle.

We now can complete the proof of Theorem 21 on the bandwidth of permutations in Im(T^m).

Proof of Theorem 21: We notice that, along the last (vertical) segment of any path of ρ, the corresponding path of π coincides with it (these areas are marked by yellow triangles in Figure 15). More formally, it follows from Proposition 23 and from Equations (26) (which describe the shape of the S- and L-paths of ρ) that

$$\text{for each } j\ (1 \le j \le k):\ m \ge n-k-1+j \Rightarrow s_j(\mathsf{shadow}(T^m(\pi))) = j, \tag{28}$$

$$\text{for each } j\ (1 \le j \le n-k):\ m \ge k-1+j \Rightarrow \ell_j(\mathsf{shadow}(T^m(\pi))) = n+1-j. \tag{29}$$

Now, we fix m ($0 \le m \le n-1$), and let $\tau = T^m(\pi) = b_1 b_2 \dots b_n$. Since $s_j(\mathsf{shadow}(\tau)) = \alpha$ for some j implies $b_\alpha \le k$, and since $\ell_j(\mathsf{shadow}(\tau)) = \beta$ for some j implies $b_\beta \ge k+1$, equations (28) and (29) translate to

$$1 \le i \le k+m+1-n \quad \Rightarrow \quad b_i \le k, \tag{30}$$

$$k+n-m \le i \le n \quad \Rightarrow \quad b_i \ge k+1. \tag{31}$$

Now we make the final step. Let $1 \le i \le n$. Equations (30) and (31) hold for *any* k, $1 \le k \le n-1$, and we choose two specific values. First, we take $k = i+n-1-m$. Now we have $i \le k+m+1-n$, and therefore (30) implies $b_i \le k = i+(n-1-m)$. Second, we take $k = i-n+m$. Now we have $i \ge k+n-m$, and therefore (31) implies $b_i \ge k+1 = i-(n-1-m)$. Thus, we have $|b_i - i| \le n-1-m$ for each i ($1 \le i \le n$), that is, $\mathsf{d}(\tau) \le n-1-m$, as claimed. □

In terms of permutation diagrams, Equations (30) and (31) (for fixed m) mean that for each k some rectangular areas in the diagram of $\tau \in \mathrm{Im}(T^m)$ are forbidden. Taking the union of these areas for $1 \le k \le n-1$, we obtain **forbidden corners**, which yield the bound on the bandwidth. See Figure 16 for illustration (the forbidden areas are indicated by grey colour).

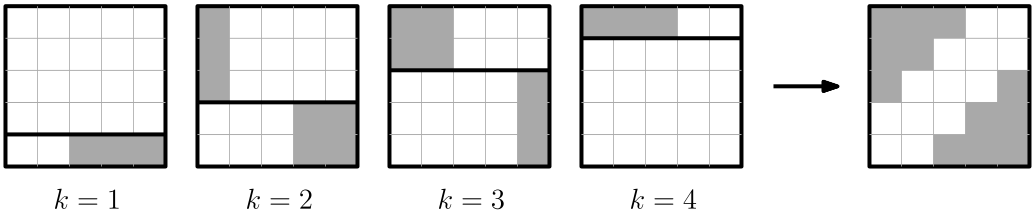


Fig. 16: Illustration to proof of Theorem 21 for $n=5$: forbidden areas for fixed m ($m=3$).

In Figure 17 we show the forbidden corners for $n=5$ and $0 \le m \le 4$.

Now, if we compare this to Figure 3 (where grey areas are precisely those forbidden by Theorem 21), we see many other areas without points, and it is natural to ask whether there are some larger forbidden areas. In fact, contrary to what Figure 3 could suggest, the forbidden areas characterized in Theorem 21 are maximal in the sense that for each position outside the grey corners we can find a permutation that contains a point at this position. In fact, all the positions not forbidden by Theorem 21 are "covered" by skew-layered permutations with ≤ 2 runs, as our next result shows.

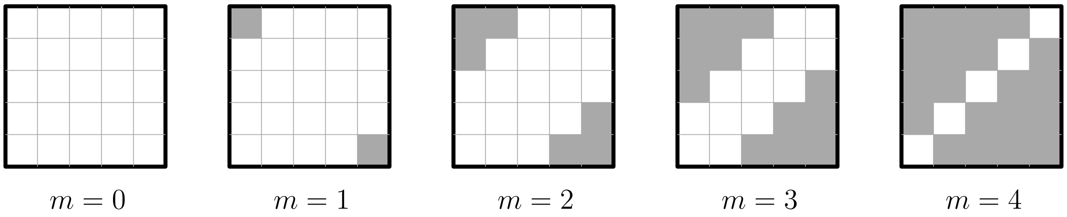


Fig. 17: Illustration to proof of Theorem 21 for $n = 5$: forbidden areas for $0 \leq m \leq n-1$.

Proposition 24. *Let n be fixed, and let $0 \leq m \leq n-1$. For $1 \leq i \leq n$, let $\max\{0, i-(n-1-m)\} \leq j \leq \min\{n, i+(n-1-m)\}$. Then there exists π, a skew-layered permutation with ≤ 2 runs, such that for $\tau = T^m(\pi) = b_1 b_2 \dots b_n$ we have $b_i = j$.*

Proof: Assume $j < i$. Let $\pi = \ominus(k, n-k)$ where k will be determined below. From (26) and discussion thereafter, we know: if $m \leq j-1$ then $j = b_i$ for $i = k+j$, if $j-1 \leq m \leq k-1+j$ then $j = b_i$ for $i = k+j-(m-(j-1))$, and if $k-1+j \leq m \leq n-1$ then $j = b_j$. Solving the first two expressions for k we obtain that we need to set $k = i-j$ if $m \leq j-1$, and $k = i-2j+m+1$ if $j \leq m \leq k-1+j$.

The case $j > i$ is similar, and for $j = i$ we can take $\pi = \mathsf{id}$ which will do for any m. □

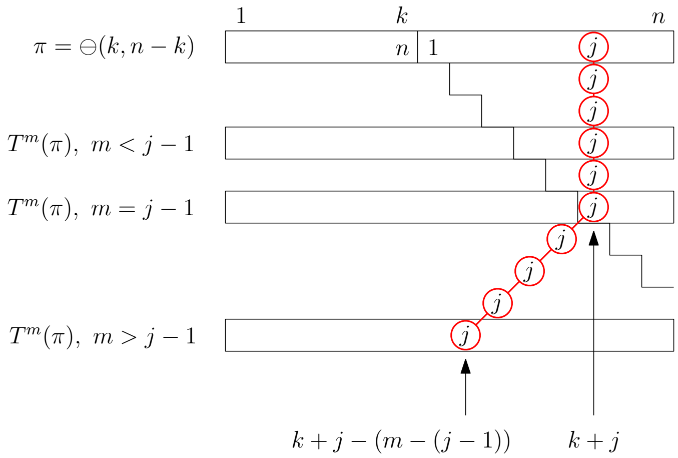


Fig. 18: Illustration to the proof of Proposition 24.

Finally, we remark that, since Theorem 21 establishes that the bandwidth of allowed positions for permutations in $\mathrm{Im}(T^m)$ decreases with m, one can be tempted to conjecture that for $\pi \neq \mathsf{id}$ we always have $\mathsf{d}(T(\pi)) < \mathsf{d}(\pi)$. Such a claim would, of course, imply Theorem 21! However, such a conjecture is wrong: the smallest counter-example is $\pi = 3412$, and it is easy to see that any skew-layered permutation with at least two blocks, such that the first and last blocks are of size ≥ 2, is a counterexample.

4.3 The penultimate step of flip-sort: the theorem about $\mathrm{Im}(T^{n-2})$.

In view of Ungar's result, we have $\mathrm{Im}(T^{n-1}) = \{\mathsf{id}\}$. The main result of this section is a structural characterization of $\mathrm{Im}(T^{n-2})$. According to Theorem 21, $\tau \in \mathrm{Im}(T^{n-2})$ implies $\mathsf{d}(\tau) \leq 1$. We will see below that $\mathsf{d}(\tau) \leq 1$ is only necessary, but not sufficient, condition for being in $\mathrm{Im}(T^{n-2})$. (For $n = 5$, one can see in Figure 2 that the permutation 21354 with $\mathsf{d} = 1$ is not in $\mathrm{Im}(T^3)$.) However, permutations that satisfy $\mathsf{d}(\tau) \leq 1$ with will play an important role in our next results, and so we introduce a notation for them, and list some of their (nearly obvious) properties.

Definition 25. A permutation τ is *thin* if $\mathsf{d}(\tau) \leq 1$.

Observation 26. *1. A thin permutation is uniquely defined by the sequence of lengths of its runs. The thin permutation with runs of lengths $r_1, r_2, \ldots, r_s$ (left to right) will be denoted by $\mathsf{thin}(r_1, r_2, \ldots, r_s)$. For example, $\mathsf{thin}(3, 4, 1) = 124|3568|7$. If $s \geq 2$, the runs r_i with $1 < i < s$ will be referred to as* inner runs.

2. *A sequence $(r_1, r_2, \ldots, r_s)$ of natural numbers is realizable as the sequence of lengths of runs of a thin permutation if and only if $r_i > 1$ for $2 \leq i \leq s - 1$.*

3. *Any inner run of a thin permutation $a_1 a_2 \ldots a_n \neq \mathsf{id}$ has the form*

$$[a_\ell, a_{\ell+1}, a_{\ell+2}, \ldots, a_{m-2}, a_{m-1}, a_m] = [\ell-1, \ell+1, \ell+2, \ldots, m-2, m-1, m+1]$$

The first and, respectively, the last run, have the forms

$$[a_1, a_2, \ldots, a_{m-2}, a_{m-1}, a_m] = [1, 2, \ldots, m-2, m-1, m+1],$$

$$[a_\ell, a_{\ell+1}, a_{\ell+2}, \ldots, a_{n-1}, a_n] = [\ell-1, \ell+1, \ell+2, \ldots, , n-1, n].$$

4. *A thin permutation is also uniquely defined by the sequence of lengths of its falls, that can only have lengths 1 or 2. Any $\{1, 2\}$-sequence is realizable as the sequence of lengths of falls of a thin permutation. It follows that thin permutations are enumerated by Fibonacci numbers.*

Now we state and prove the characterization of $\mathrm{Im}(T^{n-2})$.

Theorem 27. *A permutation $\tau = b_1 b_2 \ldots b_n$ belongs to $\mathrm{Im}(T^{n-2})$ if and only if it is thin and has no inner runs of odd size.*

Accordingly, $|\mathrm{Im}(T^{n-2})| = 2^{n/2-1} + 2^{n/2} - 1$ for even n, $2^{(n+1)/2} - 1$ for odd n (OEIS A052955).

Proof: [⇒] Let $\tau = b_1 b_2 \ldots b_n \in \mathrm{Im}(T^{n-2})$. By Theorem 21, we have $\mathsf{d}(\tau) \leq 1$. The second condition will be proved by contradiction. To this end we assume that τ is a thin permutation with at least one inner run of odd size, and that we have $\tau = T^{n-2}(\pi)$ for some permutation π. Specifically, we assume that τ has a run $[b_{k+1}, b_{k+2}, \ldots, b_{\ell-1}, b_\ell]$ such that $1 \leq k$, $\ell \leq n-1$, $\ell - k$ is odd, and $\ell - k \geq 3$ (because, by Observation 26(2), a thin permutation has no inner runs of size 1).

For the proof, we shall exploit shadows and wiring diagrams for both k and ℓ. First, due to Observation 26(3), we know the k- and the ℓ-shadows of τ:

$$\mathsf{shadow}_k(\tau) = \mathsf{S}^{k-1}\mathsf{L}\mathsf{S}\mathsf{L}^{n-k-1}, \quad \mathsf{shadow}_\ell(\tau) = \mathsf{S}^{\ell-1}\mathsf{L}\mathsf{S}\mathsf{L}^{n-\ell-1}. \tag{32}$$

Now we introduce the $(k-\ell)$-*wiring diagram of* π: the drawing that consists of S-paths from the k-wiring diagram of π, and the L-paths from the ℓ-wiring diagram of π. Thus, in the $(k-\ell)$-wiring diagram we have k (red) S-paths: for $1 \le j \le k$, the jth S-path (denoted by S_j) connects the jth from left occurrences of the values from $S := \{1, 2, \ldots, k\}$; and we have $\ell' := n - \ell$ (blue) L-paths: for $1 \le j \le \ell'$ the jth L-path (denoted by L_j) connects the jth from right occurrences of the values from $L := \{\ell+1, \ell+2, \ldots, n\}$. The remaining values from $\{k+1, k+2, \ldots, \ell\}$ are not connected by paths in such a diagram.

Additionally, we consider the pink paths of $\rho_k = \ominus(n-k, k)$ and the light-blue paths of $\rho_\ell = \ominus(n-\ell, \ell)$. By Proposition 23, these paths majorate corresponding paths of π: For $1 \le j \le k$, the path $\mathsf{S}_j(\pi)$ is weakly to the left from the $\mathsf{S}_j(\rho_k)$, and for $1 \le j \le \ell'$, the path $\mathsf{L}_j(\pi)$ is weakly to the right from the $\mathsf{L}_j(\rho_\ell)$.

At this point we notice that, **since** $\ell - k$ **is odd**, all the pink paths $\mathsf{S}_j(\rho_k)$ and the light-blue paths $\mathsf{L}_j(\rho_\ell)$ cross each other at grid points of the diagram, see Figure 19. (Compare with Figure 12, where pink and light-blue paths cross each other between the rows of the diagram!) Denote by $P_{i,j}$ ($1 \le i \le k$, $1 \le j \le \ell'$) the crossing point of the path $\mathsf{S}_i(\rho_k)$ and the path $\mathsf{L}_j(\rho_\ell)$: we refer to these points as *conflict points*.

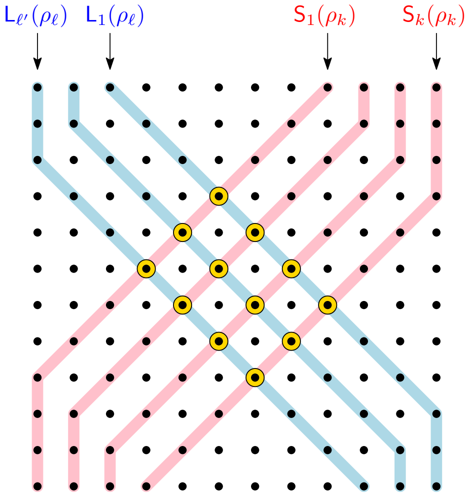


Fig. 19: Illustration to the proof of Theorem 27: pink paths of $\rho_k = \ominus(n-k, k)$ and blue-light paths of $\rho_\ell = \ominus(n-\ell, \ell)$. The conflict points are drawn as yellow discs.

Since $S \cap L = \varnothing$, it is impossible that both $\mathsf{S}_i(\pi)$ and $\mathsf{L}_j(\pi)$ pass through $P_{i,j}$. This means that in the row of $P_{i,j}$, at least one of the paths $\mathsf{S}_i(\pi)$ and $\mathsf{L}_j(\pi)$ is **strongly majorated** by the corresponding pink or, respectively, light-blue path.

For $1 \le i \le k$, $1 \le j \le \ell'$ we say that the pair (i, j) is *regular* if some point of $\mathsf{S}_i(\pi)$ lies on $\mathsf{S}_i(\rho_k)$ after $P_{i,j}$ and before $\mathsf{S}_i(\pi)$ and $\mathsf{S}_i(\rho_k)$ reach their final position i, *and* some point of $\mathsf{L}_j(\pi)$ lies on $\mathsf{L}_j(\rho_\ell)$ after $P_{i,j}$ and before $\mathsf{L}_j(\pi)$ and $\mathsf{L}_j(\rho_\ell)$ reach their final position $n+1-j$. Next we prove two claims:

- **Claim 1:** *The pair (k, ℓ') is regular.*
  Indeed, we have $b_{k+1} = k$ and $b_\ell = \ell + 1$, which means that in the row that corresponds to $\tau = T^{n-2}(\pi)$, $\mathsf{S}_k(\pi)$ coincides with $\mathsf{S}_k(\rho_k)$, and $\mathsf{L}_{\ell'}(\pi)$ coincides with $\mathsf{L}_{\ell'}(\rho_\ell)$ In both cases the paths are after the conflict points and before their final columns.

- **Claim 2:** *If a pair (i, j) is regular, then at least one of the pairs $(i-1, j)$, $(i, j-1)$ is regular.*
  At least one of the paths $\mathsf{S}_i(\pi)$, $\mathsf{L}_j(\pi)$ does not contain the point P_{ij}. Assume without loss of generality that $\mathsf{S}_i(\pi)$ does not contain the point P_{ij}. That is, in the row of P_{ij}, $\mathsf{S}_i(\pi)$ is strictly to the left from the pink path $\mathsf{S}_i(\rho_k)$. Since $\mathsf{S}_i(\pi)$ has later a point that lies on $\mathsf{S}_i(\rho_k)$ (after $P_{i,j}$, but before both paths reach their final position i), it makes a vertical step there, from a point α to a point β (refer to Figure 20). But this means that γ, the left neighbour of α, is also from S. Therefore γ is a point of $\mathsf{S}_{i-1}(\pi)$ on $\mathsf{S}_{i-1}(\rho_k)$, after $P_{i-1,j}$ but before both paths reach their final position $i-1$. This means that the pair $(i-1, j)$ is regular. (If we assume above that $\mathsf{L}_j(\pi)$ does not contain the point P_{ij}, we obtain that $(i, j-1)$ is regular.)

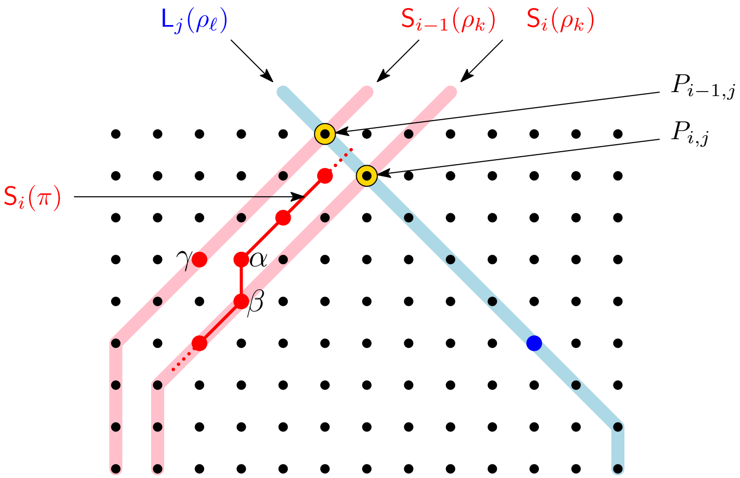


Fig. 20: Illustration to the proof of the claim: if (i, j) is regular, then $(i-1, j)$ is regular or $(i, j-1)$ is regular.

Starting with Claim 1 and applying Claim 2 repeatedly, we obtain that the pair $(1, 1)$ is regular. However, the proof of Claim 2 applies to $(1, 1)$ as well. Since $(1, 1)$ is the highest conflict point, we get a contradiction.

It is also possible to reverse this proof and to explain it "from the top to the bottom". Then it essentially says that at least one of the sets, S and L, arrives at its final position sooner than in $n-1$ iterations. However in τ these sets are not at their final position (as witnessed by (32)), and this is a contradiction.

[$\Leftarrow$] For each τ, a thin permutation without odd inner runs, we will construct π such that $T^{n-2}(\pi) = \tau$. If $\tau = \mathsf{id}$ we can take $\pi = \mathsf{id}$, so we assume from now on that $\tau \neq \mathsf{id}$. Let $\tau = \mathsf{thin}(r_1, r_2, \ldots, r_{s-1}, r_s)$ so that $s > 1$ and the numbers $r_2, \ldots, r_{s-1}$ are even. We define π to be the skew-layered permutation $\ominus(r_s, r_{s-1}, \ldots, r_2, r_1)$.

We claim that for this π we have $T^{n-2}(\pi) = \tau$. To see that, we analyse the wiring diagram of $\pi, T(\pi), T^2(\pi), \ldots$; an example is illustrated in Figure 21. Notice that this is the usual wiring diagram, in which paths connect fixed values. For the sake of visualization, each inner run of π is partitioned into two *blocks* on equal size, the left one and the right one. For the values from the left blocks of inner runs, and for all the values from the last run, we colour the paths in green. For the values from the right blocks of inner runs, and for all the values from the first run, we colour the paths in orange. Then, the paths have very clear description. Specifically, for an inner run of length r, the jth from left value ($1 \le j \le r/2$) generates the green path that consists of 5 segments (some of them can be empty): (1) a vertical segment of length $(j-1)$, (2) a slanted segment of slope 1 until the path reaches the position j, (3) a vertical segment of length $r - 2j + 1$, (4) a slanted segment of slope -1 until the path reaches its final position, (5) a vertical segment. The green paths of the last run do not have (4) and (5); orange paths are described similarly. The whole diagram is symmetric, and therefore the sequence of lengths of runs of τ is the reversal of the sequence of lengths of runs of π.

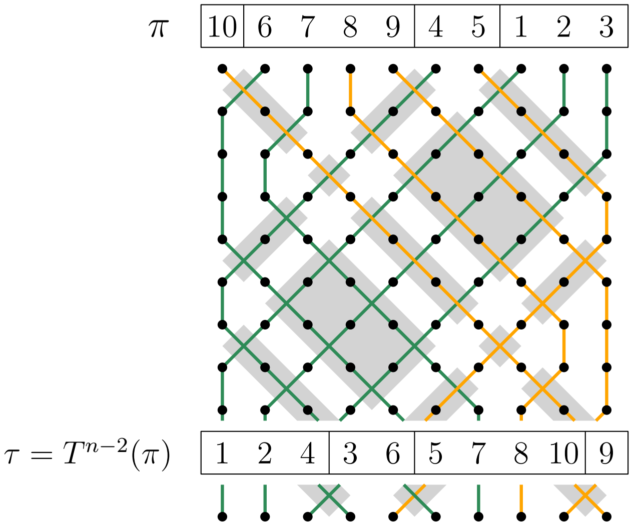


Fig. 21: Illustration to the proof of $T^{n-2}(\ominus(r_s, \ldots, r_1)) = \mathsf{thin}(r_1, \ldots, r_s)$ (where $r_2, \ldots, r_{s-1}$ are even).

Our proof shows that in order to construct a permutation in $\mathrm{Im}(T^{n-2})$, one only has to choose positions (of the same parity) for the gaps between runs. This yields the enumerative formula, as stated in the claim. □

Remark. In general, there are several permutations π that satisfy $T^{n-2}(\pi) = \tau$ for given $\tau \in \mathrm{Im}(T^{n-2})$. Characterizing all such π is a challenging problem.

4.4 Skew-layered permutations: a family of worst cases

By Theorem 27, each skew-layered permutation of size n without odd inner runs has the (maximum possible for permutations of size n) cost $n-1$. Characterizing all permutations with that cost is a challenging open problem. In this section, we present a necessary condition, and then we conclude with a conjecture concerning the cost of any skew-layered permutation.

Proposition 28. *Let π be a permutation of size n with $\mathsf{cost}(\pi) = n-1$. Let $\tau = b_1 b_2 \dots b_n = T^{n-2}(\pi)$.*

1. *There exists k, $1 \le k \le n-1$, such that $\mathsf{shadow}_k(\tau) \neq \mathsf{S}^k\mathsf{L}^{n-k}$.*
2. *For each such k, we have $\mathsf{shadow}_k(\pi) = \mathsf{L}^{n-k}\mathsf{S}^k$. Consequently, π can be written as a skew sum $\sigma \ominus \sigma'$, where σ, σ' are permutations of sizes $n-k, k$.*

Proof:

1. Since $\mathsf{cost}(\pi) = n-1$, we have $\tau \neq \mathsf{id}$. For each i with $b_i > i$ we can take $k = b_i - 1$, and for each i with $b_i < i$ we can take $k = b_i$.
2. In order to find a contradiction, assume that $\mathsf{shadow}_k(\pi) \neq \mathsf{L}^{n-k}\mathsf{S}^k$. Since the only element of the set of words $\mathcal{SL}(n-k,k)$ covered by $\mathsf{L}^{n-k}\mathsf{S}^k$ is $\mathsf{shadow}_k(T(\rho)) = \mathsf{L}^{n-k-1}\mathsf{SLS}^{k-1}$, we necessarily have $\mathsf{shadow}_k(\pi) \preccurlyeq \mathsf{shadow}_k(T(\rho))$. Then, by Proposition 22 and by induction, we have $\mathsf{shadow}_k(T^{n-2}(\pi)) \preccurlyeq \mathsf{shadow}_k(T^{n-1}(\rho))$, that is, $\mathsf{shadow}_k(\tau) = \mathsf{S}^k\mathsf{L}^{n-k}$, which is a contradiction.

□

As for skew-layered permutations, we expect that their cost is always high (excluding the cases of id and $-\mathsf{id}$), however it is not always $n-1$. Our computer experiments then suggest the following classification.

Conjecture 29. *Let π be a skew-layered permutation of size n such that π is neither id nor $-\mathsf{id}$.*

- *For even n, one has $\mathsf{cost}(\pi) = n-1$.*
- *For odd n, if the $(n+1)/2$-st position of π is the central point of a run or of a fall of size ≥ 3, then $\mathsf{cost}(\pi) = n-2$. Otherwise, $\mathsf{cost}(\pi) = n-1$.*

As a direct corollary of this conjecture, among the 2^{n-1} skew-layered permutations, there are:

- for even n, $2^{n-1} - 2$ skew-layered permutations with $\mathsf{cost}(\pi) = n-1$;
- for odd n, $(2^{n-2} - 2)/3$ skew-layered permutations with $\mathsf{cost}(\pi) = n-2$,
- for odd n, $(5 \cdot 2^{n-2} - 4)/3$ skew-layered permutations with $\mathsf{cost}(\pi) = n-1$.

These are respectively the sequences `A000918`, `A020988`, and `A080675` from the OEIS.

5 Conclusion

In this article, we have seen that a simple sorting procedure like flip-sort has many interesting combinatorial and algorithmic facets. Our analysis of the best cases and the worst cases of this algorithm, our results on the underlying poset, the links with lattice paths, are just some first steps towards a more refined understanding of this process.

En passant, we listed a few conjectures, which, we hope, could tease the desire of the reader to also work on this topic. The optimal cost of the computation of pop-stacked permutations is still open; other challenges are a finer analysis of the automata which recognize them (when the number of runs is fixed), the existence of closed-form formulas, a method to solve puzzling functional equations like the one given in Theorem 10.

We also believe that several of the notions and structures introduced in this article can also be useful for the next natural step: the average-case analysis of flip-sort. Even if this algorithm is not, in its naive version, as efficient as quicksort, it is very easy to implement, to parallelize, and it is still possible to optimize its implementation (this is a subject per se, like using double chained lists to perform efficiently the flips, or using some additional data structures to have a parsimonious reading of each iterations, etc). So there is not a "flip-sort" algorithm, but a family of flip-sort algorithms, and to finely tune them in order to decide what could be the optimal variant should clearly be the subject of future research.

Acknowledgements

This research was supported by the Austrian Science Fund (FWF) in the framework of the project *Analytic Combinatorics: Digits, Automata and Trees* (P 28466). Cyril Banderier thanks the Department of Mathematics at the University of Klagenfurt for funding his visit in July 2019.

References

[1] Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK*. Springer, 2018. Revised and enlarged 6th edition (1st ed: 1998).

[2] Gadi Aleksandrowicz, Andrei Asinowski, and Gill Barequet. Permutations with forbidden patterns and polyominoes on a twisted cylinder of width 3. *Discrete Math.*, 313(10):1078–1086, 2013.

[3] Andrei Asinowski, Cyril Banderier, Sara Billey, Benjamin Hackl, and Svante Linusson. Pop-stack sorting and its image: permutations with overlapping runs. *Acta Math. Univ. Comenian. (N.S.)*, 88(3):395–402, 2019.

[4] Andrei Asinowski, Cyril Banderier, and Benjamin Hackl. On extremal cases of pop-stack sorting. In *Permutation Patterns (Zürich, Switzerland)*, 2019.

[5] Andrei Asinowski and Toufik Mansour. Separable d-permutations and guillotine partitions. *Ann. Comb.*, 14(1):17–43, 2010. [doi].

[6] Michael D. Atkinson and Jörg-Rüdiger Sack. Pop-stacks in parallel. *Inform. Process. Lett.*, 70(2):63–67, 1999. [doi].

[7] David Avis and Monroe Newborn. On pop-stacks in series. *Util. Math.*, 19:129–140, 1981.

[8] Cyril Banderier, Mireille Bousquet-Mélou, Alain Denise, Philippe Flajolet, Danièle Gardy, and Dominique Gouyou-Beauchamps. Generating functions for generating trees. *Discrete Math.*, 246(1-3):29–55, 2002.

[9] Cyril Banderier and Philippe Flajolet. Basic analytic combinatorics of directed lattice paths. *Theoretical Computer Science*, 281(1-2):37–80, 2002.

[10] David Bevan. Permutation patterns: basic definitions and notation, 2015. `arXiv:1506.06673`.

[11] Miklós Bóna. *Combinatorics of Permutations*. CRC Press, 2nd edition, 2012.

[12] Janusz A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. Proc. Symp. Math. Theor. Automata, New York 1962, 529–561, 1963.

[13] Anders Claesson and Bjarki Ágúst Guðmundsson. Enumerating permutations sortable by k passes through a pop-stack. *arXiv*, 2017.

[14] Anders Claesson, Bjarki Ágúst Guðmundsson, and Jay Pantone. Counting pop-stacked permutations in polynomial time. *arXiv*, 2019.

[15] Serge Dulucq, Sophie Gire, and Olivier Guibert. A combinatorial proof of J. West's conjecture. *Discrete Math.*, 187(1-3):71–96, 1998.

[16] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[17] Dominique Foata and Marcel-Paul Schützenberger. *Théorie géométrique des polynômes eulériens*. Lecture Notes in Mathematics, Vol. 138. Springer-Verlag, Berlin-New York, 1970.

[18] William H. Gates and Christos H. Papadimitriou. Bounds for sorting by prefix reversal. *Discrete Math.*, 27(1):47–57, 1979.

[19] Jacob E. Goodman and Richard Pollack. A combinatorial perspective on some problems in geometry. *Congressus Numerantium*, 32:383–394, 1981. Proceedings of the 12th Southeastern Conference on Combinatorics, Graph Theory, and Computing, March 2–5, 1981, Baton Rouge, LA.

[20] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics. A foundation for computer science*. Addison-Wesley, 1994. Second edition (1st ed: 1988).

[21] Donald E. Knuth. *The Art of Computer Programming. Vol. 1: Fundamental Algorithms*. Addison-Wesley, Reading, MA, 1998. Third edition (1st ed: 1968).

[22] Donald E. Knuth. *The Art of Computer Programming. Vol. 3: Sorting and searching*. Addison-Wesley, Reading, MA, 1998. Second edition (1st ed: 1973).

[23] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the Stanley–Wilf conjecture. *J. Combin. Theory, Ser. A*, 107(1):153–160, 2004.

[24] T. Kyle Petersen. *Eulerian numbers*. Birkhäuser/Springer, 2015. [doi].

[25] Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger. *$A = B$*. A. K. Peters, 1996.

[26] Lara Pudwell and Rebecca Smith. Two-stack-sorting with pop stacks. *Australas. J. Combin.*, 74(1):179–195, 2019.

[27] Robert Sedgewick. *Algorithms in C. Parts 1-4: Fundamentals, data structures, sorting, searching.* Addison-Wesley, 3rd edition, 1997.

[28] Yidong Sun. The statistic "number of *udu*'s" in Dyck paths. *Discrete Math.*, 287(1-3):177–186, 2004.

[29] The SageMath Developers. *SageMath Mathematics Software (Version 9.0)*, 2020. `http://www.sagemath.org`.

[30] Peter Ungar. $2N$ noncollinear points determine at least $2N$ directions. *J. Combin. Theory Ser. A*, 33(3):343–347, 1982.

[31] Julian West. Sorting twice through a stack. *Theoret. Comput. Sci.*, 117(1-2):303–313, 1993. Conference on Formal Power Series and Algebraic Combinatorics (Bordeaux, 1991).