JOURNAL OF
UNDERGRADUATE RESEARCH

A lower bound for general t-stack sortable permutations via pattern avoidance

Pranav Chinmay
University of Florida

Faculty Mentor: Miklos Bona, Department of Mathematics

Abstract

There is no formula for general *t*-stack sortable permutations. Thus, they are studied by establishing lower and upper bounds. Permutations that avoid certain pattern sets provide natural lower bounds. This paper presents a recurrence relation that counts the number of permutations that avoid the set *(23451,24351,32451,34251,42351,43251)*. This establishes a lower bound on 3-stack sortable permutations. Additionally, the proof generalizes to provide lower bounds for all *t*-stack sortable permutations.

Keywords: pattern avoidance, stack sortability, combinatorics

Definitions and Preliminaries

Let *[n]* denote the set *{1, ... , n}*, with $n \geq 1$. An *n*-permutation is a bijection from *[n]* to *[n]*.

In this paper, permutations are written in one-line notation. For instance, the 4-permutation that sends 1 to 2, 2 to 4, 3 to 3 and 4 to 1 is written *2431* (read two-four-three-one).

Thus, one may speak of the generic *n*-permutation *p* as a sequence (□$_{□}$)as *i* runs from 1 to *n*. Accordingly, one may speak of an *m*-subsequence □′ of a permutation *p*. An *m*-subsequence is obtained by deleting (□ − □)elements from the *n*-permutation without modifying the order. For instance, *231* is a 3-subsequence of *2431*, as is *431*. *41* is a 2-subsequence of *2431*.

Next, an *m*-subsequence □′ can be *normalized* by assigning it to an *m*-permutation via the natural order-preserving bijection on the elements. For example, normalizing *431* yields *321* and normalizing *41* yields *21*. Normalizing *5389* yields *2134*. Observe that the relative ordering is preserved.

This allows a definition of the concept of pattern avoidance. A permutation *p contains* a pattern *q* if there exists a subsequence of *p* which yields *q* when normalized. For instance, *3421* contains the pattern *231*, since the subsequence *342*, when normalized, yields *231*. Conversely, a permutation *p avoids q* if it does not contain *q*. For instance, *1432* avoids *231*. None of the 3-subsequences of *1432* yield *231* when normalized: *143* yields *132*, *142* yields *132*, *132* yields *132*, and *432* yields *321*. This can be generalized by saying *p* avoids a set of patterns *S* if it avoids every pattern in *S*. Note that a subsequence avoids some given pattern if and only if its normalization does.

The next concept that can be described is that of a stack sortable permutation. Given an *n*-permutation *p*, it is possible to write *p=LnR*, where *n* is the largest element in the permutation, *L* is the subsequence on the left of *n*, *R* is the subsequence on the right of *n*, and *L*, *n*, and *R* are concatenated. Then, define the stack sorting map as the recursive function $s(p) = s(L)s(R)n$, again, with concatenation. If *s(p)* is the identity permutation *123...n*, the permutation *p* is said to be 1-stack sortable. Generalizing, a permutation *p* is said to be *t*-stack sortable if $s^t(p)$is the identity permutation.

Deducing efficient ways to sort a list of *n* objects is an important problem in algorithm design, and computer science in general. The best kind of generic sorting algorithm requires O(*n log n*) steps. However, if additional information about the list is known, more efficient algorithms may be employed. Several such algorithms were described by Donald Knuth in his seminal work The Art of Computer Programming. One of these was adopted by Julian West, who reformulated it as the modern definition of the stack sorting map.

In his dissertation, West noted that the number of 1-stack sortable *n*-permutations is given by the n^{th} Catalan number, as is the number of *n*-permutations that avoid *231*. Additionally, he conjectured a closed formula for the number of 2-stack sortable *n*-permutations. This formula was later proved by Doron Zeilberger. However, there is no conjecture for a closed formula for 3-stack-sortable *n*-permutations and beyond.

Without a closed formula, the best way to estimate the number of general *t*-stack sortable *n*-permutations is to find lower and upper bounds. The connection with pattern avoidance bears fruit to this end. A permutation *p* is 1-stack sortable if and only if it avoids *231*. It turns out that a permutation *p* is 2-stack sortable if it avoids the set $S_2 = \{2341,3241\}$, and 3-stack sortable if it avoids the set $S_3 = \{23451,24351,32451,34251,42351,43251\}$. Thus, counting the number of *n*-permutations that avoid the set S_3 provides a natural lower bound for the number of 3-stack sortable *n*-permutations.

Result

Theorem. Let $\square_{\square}$ be the number of *n*-permutations that avoid S_3. Then for $n > 4$,

$$\square_{\square} = \square\square_{\square-\square} + \frac{\square}{\square}\sum_{\square=\square}^{\square-\square} \square_{\square}\square_{\square-\square+\square}$$

with initial terms $a_1 = 1, a_2 = 2, a_3 = 6, a_4 = 24$.

Proof. If n = 1, 2, 3 or 4, the values of a_n are easy to check by inspection. So assume $n > 4$. To begin, call the patterns in the set S_3 bad patterns. An element in a permutation is *involved* in a bad pattern, or *plays a role* in a bad pattern, if it is part of a subsequence that normalizes to a bad pattern. This proof will examine the allowed positions of the numbers 1 to *n* in a series of steps, ensuring that no bad patterns occur at each step.

Step I

First, consider the largest number, *n*. Note that the only possible role the number *n* could play in a bad pattern is the role of the peak (i.e. the role of *5*).

There are *n* possible positions for the number *n* in an *n*-permutation. Look at each of these positions and count their contribution to a_n. Consider the cases where the number *n* is in position 1, 2, 3 or *n*. In each of these cases, *n* cannot be involved in a bad pattern, since its position prevents it from being a peak. Thus, in each of these cases, the contribution to a_n is precisely a_{n-1} (no more by definition, and no less since *n* cannot be involved in a bad pattern). This immediately gives a contribution of $4a_{n-1}$.

Step II

Claim: Let *p* be an *n*-permutation, and suppose *n* is in position *i*, where $4 \leq i \leq n-1$. Then the number *n* is not involved in any bad pattern if and only if the numbers 1, 2, 3, … , *i-3* are all to the left of n.

Justification: Suppose not all these numbers are to the left of n. Then certainly one of these numbers must be to the right of n. Call it *m*. Let $B = \{r \mid m < r < n\}$. Note that $|B| \geq n-i+2 \geq 3$. Since the number *n* is in position *i*, there are *i-1* spots to the left of *n* and *n-i* spots to the right of *n*. But one of the spots on the right is taken by *m*, so there are only *n-i-1* possible spots to the right of *n*. By the Pigeonhole Principle, at least three of the numbers in *B* must be to the left of *n*. This creates a bad pattern involving *n* (i.e. the three numbers in B, followed by *n*, followed by *m*).

The other direction now follows. Suppose the numbers 1, 2, 3, … , *i-3* are all to the left of *n*. As mentioned earlier, the only role *n* can play is the role of the peak. Thus, if a bad pattern were to involve *n*, it would have to involve three numbers to the left of n, and one number to the right of *n*. First, observe that every number in 1, 2, … , *i-3* is smaller than every number to the right of *n*. Thus, they cannot be involved in bad patterns that also involve any numbers to the right of *n*. Next, notice that there are only two numbers (call them *r* and *s*) not in 1, 2, … , *i-3* that live on the left of *n* (since there are only *i-1* spots to the left of n). This is too few numbers to form a bad pattern that involves *n*. Thus, there are no bad patterns involving *n*.

Now that the claim is established, the next argument follows. Fix *n* in position *i*. The only permutations that should be counted are those that do not contain any bad patterns. So it is necessary that *n* is not involved in a bad pattern. To ensure this condition is met, 1, 2, … , *i-3* must all be to the left of n. However, this does not suffice. Bad patterns might still occur without involving *n*. Where could these occur?

Step III

Observe that the only way bad patterns could occur is one of the two following situations. Either the bad pattern involves numbers that live exclusively on the left of *n*, or it occurs in a subsequence of *r*, *s* and numbers to the right of *n* (of course the bad pattern could involve only numbers on the right, but such a situation is included in the latter condition). This is because, as observed before, no number in 1, 2, … , *i-3* can be involved in a bad pattern that also involves any number to the right of n.

Thus, consider the following two subsequences: *x*, which is the subsequence containing the first *i-1* terms of the permutation, and *y*, which is the subsequence containing *r*, *s*, and then the last *n-i* terms of the permutation. Observe that *x* and *y* have two elements in common, precisely *r* and *s*. If *r* and *s* are in the same order in both *x* and *y*, then the pair *(x,y)* refers to a valid permutation. This happens exactly half the time. Thus, when *n* is in position *i*, the contribution to a_n must be $\frac{1}{2}a_{i-1}a_{n-i+2}$. To clarify, this is because *x* must contain no bad patterns (the a_{i-1} factor), *y* must contain no bad patterns (the a_{n-i+2} factor), and *r* and *s* must be in the same order in both *x* and *y*, which gives the half factor.

Summing over *i*, as *i* runs through $4 \leq i \leq n-1$, and re-indexing,

$$\square_{\square} = \square\square_{\square-\square} + \frac{\square}{\square}\sum_{\square=\square}^{\square-\square} \square_{\square}\square_{\square-\square+\square}$$

Generally speaking, let S_t be the set of patterns of length $t+2$ whose last two entries are $t+2$ and then 1. A permutation is t-stack sortable if it avoids S_t. The proof above generalizes to any such S_t. Thus, one obtains a family of recurrences that provide lower bounds for general t-stack sortable permutations. For some fixed t, the recurrence is as follows for $n > t+1$, with the first $t+1$ values of a_n simply being $n!$.

$$\square_{\square} = (\square + 1)\square_{\square-1} + \frac{1}{(\square - 1)!}\sum_{\square=\square}^{\square-2} \square_{\square}\square_{\square-\square+\square-2}$$