Sorting Permutations with Finite Depth Stacks

Timothy Goodrich, Drew Groth, Lauren Knop,
William Olson, Lara Pudwell [*], Ruyue (Julia) Yuan

Department of Mathematics and Computer Science
Valparaiso University
Valparaiso, IN 46383, USA

August 28, 2013

Abstract

In this paper we consider permutations that are West k-stack-sortable through a stack of finite depth d. We completely characterize and enumerate those permutations that are 1-stack-sortable through a stack of any fixed depth d, those that are k-stack-sortable through a stack of depth 2, and those that are 2-stack-sortable through a stack of depth 3.

1 Introduction

A permutation $\pi = \pi_1 \cdots \pi_n$ of length n is a string of n integers such that $\{\pi_1, \ldots \pi_n\} = \{1, \ldots, n\}$. Let $\mathcal{S}_n$ be the set of permutations of length n. Given $\pi \in \mathcal{S}_n$ and $\rho \in \mathcal{S}_m$ we say that π *contains* ρ as a (classical) pattern if there exist indices $1 \leq i_1 < \cdots < i_m \leq n$ such that $\pi_{i_a} < \pi_{i_b}$ if and only if $\rho_a < \rho_b$. In this case we say that $\pi_{i_1} \cdots \pi_{i_m}$ is *order-isomorphic* to ρ and write $\pi_{i_1} \cdots \pi_{i_m} \sim \rho$. If π does not contain ρ, then π is said to *avoid* ρ. For example, $\pi = 1734625$ contains $\rho = 312$, as evidenced by the subpermutation 745. On the other hand, π avoids $\rho^* = 4321$ since π has no decreasing subsequence

[*]Corresponding author

of length 4. Pattern-avoiding permutations appear naturally in a variety of contexts, including the analysis of sorting algorithms.

A *stack* is a first-in-last-out data structure that can store the elements of a permutation. Elements of the permutation may be *pushed* one at a time from the front of the permutation onto the top of the stack or they may be *popped* one at a time from the top of the stack to the end of a new output permutation. If our goal is to sort a permutation π into the increasing permutation by one pass through a stack, then stack-sorting is a deterministic procedure, using the following two criteria:

- If the first element in the input is smaller than the top element of the stack, we push the smaller element onto the stack.
- If the first element is larger than the top element of the stack, we pop the smaller element off the stack.

Let $I_n = 12\cdots n$ and $J_n = n(n-1)\cdots 1$. If $\pi \in \mathcal{S}_n$ can be transformed to I_n via one pass through a stack using this procedure, then π is said to be *1-stack-sortable.*

Two examples of sorting permutations through a stack are given in Figure 1 and Figure 2. In both figures, the input permutation is shown above the stack and the output is below the stack. Figure 1 demonstrates that the permutation 15423 is 1-stack-sortable, whereas Figure 2 shows that the permutation 15342 cannot be sorted after only one pass through a stack. In the 1970s, Knuth completely characterized which permutations can be sorted after one pass through a stack.

Theorem 1 (Knuth [3])**.** *$\pi \in \mathcal{S}_n$ is 1-stack-sortable if and only if π avoids the pattern 231.*

As expected, 15423 avoids 231, whereas 15342 contains 231 via the subsequence 342. Indeed, the difficulty in sorting 15342 occurs when 3 is at the top of the stack and 4 is awaiting input. Popping the 3 from the stack leaves 3 before 2 in the output; pushing 4 onto the stack puts 4 before 3 in the output, and thus the permutation is unsortable.

It is well-know that the number of permutations avoiding 231 is given by $\frac{\binom{2n}{n}}{n+1}$, i.e. the nth Catalan number. Rotem [6] used a well-known correspondence between 231-avoiding permutations and binary trees to analyze monotonic subsequences within these stack-sortable permutations.

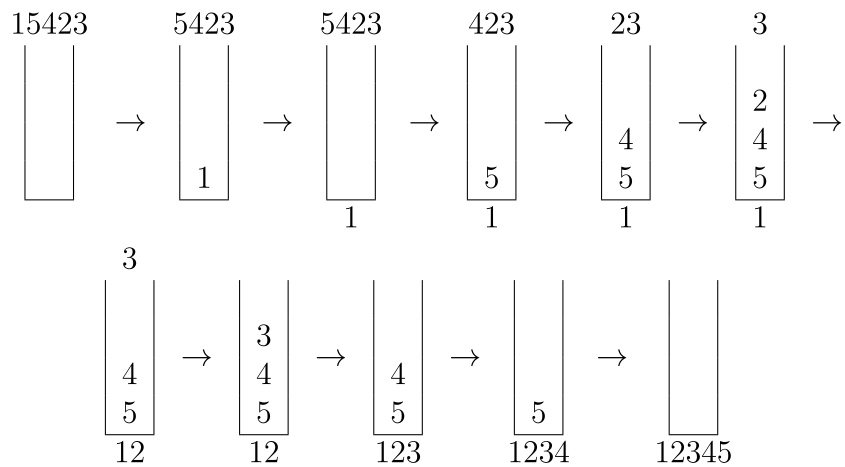

Figure 1: Sorting 15423

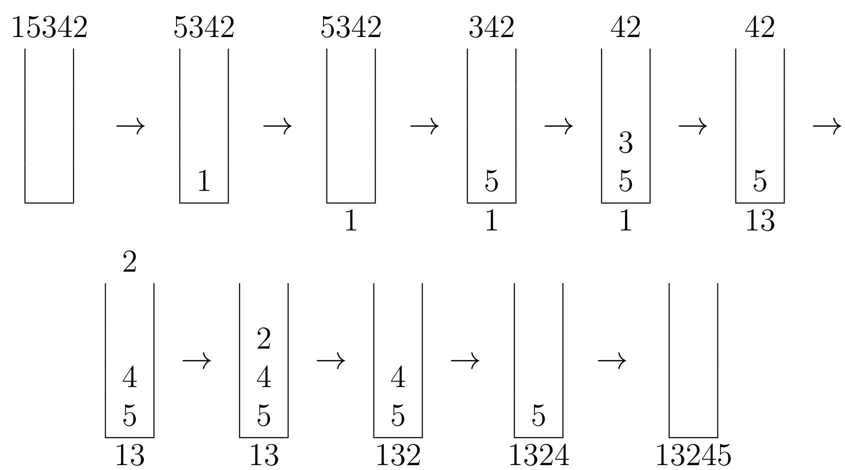

Figure 2: Sorting 15342

Now, more generally, let $S(\pi)$ be the result of passing $\pi \in \mathcal{S}_n$ through a stack one time. If $S^k(\pi) = I_n$, π is said to be *k-stack-sortable*. For example, $S(15342) = 13245$, but $S^2(15342) = S(S(15342)) = S(13245) = 12345$, so 15342 is 2-stack-sortable. West characterized the 2-stack-sortable permutations as follows.

Theorem 2 (West [11, 12]). *π is 2-stack-sortable if and only if π avoids the pattern 2341 and π contains no copy of 3241 without a larger element between the first two elements.*

The restriction on copies of 3241 requires further notation. To that end π is said to avoid the barred pattern $3\bar{5}241$ if every copy of 3241 extends to a copy of 35241; thus 2-stack-sortable permutations are those that avoid 2341 and $3\bar{5}241$. West conjectured an enumeration for these 2-stack-sortable permutations which was later proven analytically by Zeilberger [13] and combinatorially by Dulucq, Gire, and Guibert [2].

Theorem 3. *The number of 2-stack-sortable permutations is given by*

$$\frac{2(3n)!}{(n+1)!(2n+1)!}.$$

In 2012, Ulfarsson [9] characterized 3-stack-sortable permutations as those avoiding a specific set of 6 classical patterns and 4 decorated patterns. The enumeration of these permutations is open. Characterization and enumeration of k-stack-sortable permutations for $k \geq 4$ also remains open.

Thus far, all our stacks have been of arbitrary depth. For the rest of this paper, we consider what happens to the above results when we restrict the depth of our stack to a fixed size d. As we saw in Figure 1, the permutation 15423 is 1-stack-sortable in a stack of infinite depth. Figure 3 shows the result of sorting 15423 once through a depth 2 stack. When the stack is at capacity, the only choice is to pop the top element to output; when the stack has 1 or 0 elements we use the sorting procedure given above.

Let $S_d(\pi)$ be the result of sorting π once through a depth d stack. Figure 3 shows that although $S(15423) = 12345$, $S_2(15423) = 14235$. However, $S_2^2(15423) = S_2(S_2(15423)) = S_2(14235) = 12345$, so 15423 is 2-stack-sortable in a depth 2 stack.

We now turn our attention to the characterization and enumeration of those permutations of length n that are k-stack-sortable in a depth d stack. In

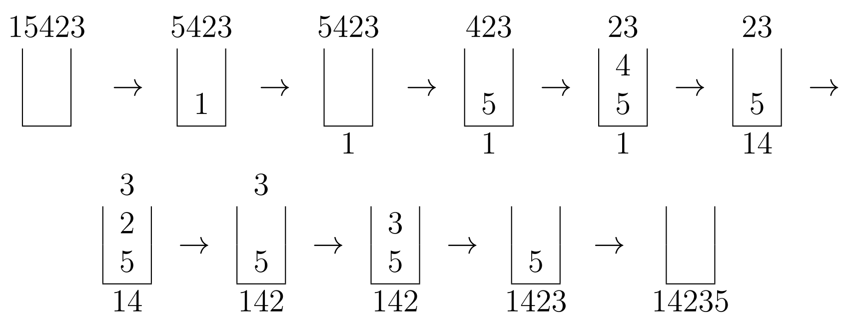

Figure 3: Sorting 15423 in a depth 2 stack

Section 2 we complete this characterization for 1-stack-sortable permutations with any depth $d \geq 1$ stack, and in Section 3 we do the same for k-stack-sortable permutations ($k \geq 1$) when $d = 2$. In Section 4, we characterize and enumerate 2-stack-sortable permutations in a depth 3 stack.

2 1-Stack-Sortable Permutations in a Depth d Stack

We first consider those permutations $\pi \in \mathcal{S}_n$ for which $S_d(\pi) = I_n$. As it turns out, avoiding 231 is still a necessary condition for sortability, but it is no longer sufficient since the limited depth of the stack may also prevent a permutation from being sorted.

Theorem 4. *$\pi \in \mathcal{S}_n$ is 1-stack-sortable in a depth d ($d \geq 1$) stack if and only if π avoids the patterns 231 and J_{d+1}.*

We prove each direction of this theorem separately.

Lemma 5. *If $\pi \in \mathcal{S}_n$ contains 231 or J_{d+1}, then π is not 1-stack-sortable in a depth d stack ($d \geq 1$).*

Proof. Suppose π contains J_{d+1}. Then there exist $\pi_{i_1}\pi_{i_2}\cdots\pi_{i_{d+1}}$ where $\pi_{i_1} > \pi_{i_2} > \cdots > \pi_{i_{d+1}}$ and $i_1 < i_2 < \cdots < i_{d+1}$. If π is sortable, $\pi_{i_{d+1}}$ exits the stack before $\pi_{i_1}, \ldots, \pi_{i_d}$. However, $\pi_{i_1}, \ldots, \pi_{i_d}$ enter the stack before $\pi_{i_{d+1}}$ and fill the stack to capacity before $\pi_{i_{d+1}}$ can be pushed in. Thus, π_{i_d} is popped from the stack before $\pi_{i_{d+1}}$, rendering π unsortable after only one pass. Therefore, if π contains J_{d+1}, π is not sortable after one pass through a depth d stack.

Next, suppose π contains a 231 pattern. If π is sortable by a depth d stack, then π is sortable by an arbitrary depth stack. By the contrapositive of this, we are done. $\square$

Now that we have seen that avoiding 231 and J_{d+1} are necessary conditions, we show that avoiding them is sufficient to be 1-stack-sortable in a depth d stack.

Lemma 6. *Suppose that $\pi \in \mathcal{S}_n$ avoids both 231 and J_{d+1}. Then π is 1-stack-sortable in a depth d stack ($d \geq 1$).*

Proof. We prove this lemma by induction on both n and d.

When $d = 1$, we wish to show that permutations avoiding 231 and 21 are sortable in a depth 1 stack. However, the only 21-avoiding permutation of length n is the increasing permutation, which is clearly sortable.

When $n = 1$, we note that the only permutation in $\mathcal{S}_1$ is 1, which avoids both given patterns and is indeed 1-stack-sortable for any $d \geq 1$.

Now, consider permutation $\pi \in \mathcal{S}_{n^*}$, which avoids 231 and J_{d^*+1} and is to be sorted in a depth d^* stack. Suppose the lemma holds if $d < d^*$ or if $n < n^*$. Let $\pi_m = n^*$, the maximum element of π, and let $\pi^L = \pi_1 \cdots \pi_{m-1}$ and $\pi^R = \pi_{m+1} \cdots \pi_{n^*}$. Since π avoids 231, all elements of π^L are strictly less than all elements of π^R. Note that when π_m enters the stack, all elements of π^L must have been popped to output because they are all smaller than π_m. Further, because π_m is larger than all elements of π^R, π_m will sit in the bottom of the stack while π^R is sorted, effectively requiring π^R to be sorted through a depth $d^* - 1$ stack. Thus, $S_{d^*}(\pi) = S_{d^*}(\pi^L)S_{d^*-1}(\pi^R)\pi_m$. We wish to show that $S_{d^*}(\pi^L)S_{d^*-1}(\pi^R)\pi_m = 1 \cdots n^*$.

Since π avoids 231 and J_{d^*+1}, so does π^L. Since $\left|\pi^L\right| < n^*$, by the induction hypothesis, $S_{d^*}(\pi^L) = 1 \cdots (m-1)$. Also, since π avoids 231 and J_{d^*+1}, π^R avoids 231 and J_{d^*}, which means $S_{d^*-1}(\pi^R) = m \cdots (n^* - 1)$. Together, we have that $S_d^*(\pi^L)S_{d^*-1}(\pi^R)\pi_m = 1 \cdots n^*$. $\square$

Magnusson and Ulfarsson [5, Proposition 3.2] achieved this characterization result independently using a sorting pre-image algorithm. Permutations avoiding 231 and a decreasing pattern have also been enumerated previously. In fact, Dairyko, Pudwell, Tyner, and Wynn [1] showed such permutations to be in bijection with certain pattern-avoiding binary trees and gave the generating function below.

$d\backslash n$	1	2	3	4	5	6	7	8	Gen. Func.	OEIS number
1	1	1	1	1	1	1	1	1	$\frac{x}{1-x}$	trivial
2	1	1	2	4	8	16	32	64	$\frac{x-x^2}{1-2x}$	A000079
3	1	1	2	5	13	34	89	233	$\frac{x-2x^2}{1-3x+x^2}$	A001519
4	1	1	2	5	14	41	122	365	$\frac{x-3x^2+x^3}{1-4x+3x^2}$	A007051
5	1	1	2	5	14	42	131	417	$\frac{x-4x^2+3x^3}{1-5x+6x^2-x^3}$	A080937
6	1	1	2	5	14	42	132	428	$\frac{x-5x^2+6x^3-x^4}{1-6x+10x^2-4x^3}$	A024175

Table 1: Enumeration of 1-stack-sortable permutations in a depth $d \leq 6$ stack

Theorem 7. *Let $d \in \mathbb{Z}^+$ and let $s_{n,d}$ be the number of permutations of length n sortable after one pass through a depth d stack. Then,*

$$\sum_{n\geq 1} s_{n,d}x^n = \frac{\sum_{i=0}^{\lfloor \frac{d}{2} \rfloor}(-1)^i \cdot \binom{d-i}{i} \cdot x^{i+1}}{\sum_{i=0}^{\lfloor \frac{d+1}{2} \rfloor}(-1)^i \cdot \binom{d+1-i}{i} \cdot x^i}.$$

For $d \leq 6$, Table 1 gives values of $s_{n,d}$ for small n, $\sum_{n\geq 1} s_{n,d}x^n$ for fixed d, and the corresponding entry in the Online Encyclopedia of Integer Sequences [7]. A few additional patterns in this data are worth mentioning.

- When $d \geq n$, $s_{n,d} = C_n$, the nth Catalan number. This is expected because the stack depth is sufficiently large to make no difference in which permutations are sortable.

- $s_{n,n-1} = C_n - 1$ because J_n is the only permutation that is sortable in a depth n stack but not in a depth $n-1$ stack.

- $s_{n,n-2} = C_n - (2n-2)$. There are $2n-2$ permutations that are sortable in a depth n stack but not in a depth $n-2$ stack. These are the permutations of length n that avoid 231 and contain J_{n-1}. There are n such permutations where $2, \ldots, n$ are in decreasing order and 1 is inserted somewhere. There are n such permutations where $\{1, \ldots, n\} \setminus \{i\}$ are in decreasing order with element i in position $n-1$. However, we have just double-counted $n(n-1)\cdots 4312$ and $n(n-1)\cdots 4321$ yielding $2n-2$ permutations.

- Apparently, $s_{n,n-k}$ has the form $C_n - f(n)$ where $f(n)$ is a polynomial of degree $k-1$.

3 k-Stack-Sortable Permutations in a Depth 2 Stack

In the general case of sorting with an infinite-depth stack, only classical permutation patterns were required to characterize 1-stack-sortable permutations, so it is unsurprising that the characterization in Theorem 4 uses only classical patterns. We now consider another case of stack-sortability that only requires classical permutation patterns.

Theorem 8. *$\pi \in \mathcal{S}_n$ is k-stack-sortable in a depth 2 stack if and only if π avoids $\{\rho \in \mathcal{S}_{k+2} | \rho_{k+2} = 1\}$.*

Proof. First, note that when π contains a pattern of length $k+2$ ending in 1, the smallest element in this pattern can only move ahead of one other element from the pattern per sorting iteration. Therefore, it takes at least $k+1$ sorts to sort π.

On the other hand, suppose that $\pi \in \mathcal{S}_{n^*}$ avoids $\{\rho \in \mathcal{S}_{k^*+2} | \rho_{k^*+2} = 1\}$ and is to be sorted k^* times in a depth 2 stack and that the theorem holds when $n < n^*$ or $k < k^*$.

We have shown the theorem is true for all $n \geq 0$ when $k = 1$, and the theorem is trivially true for all $k \geq 0$ when $n = 0$ or $n = 1$. We proceed by double induction.

Let $\pi_m = n^*$, the maximum element of π. Let $\pi^L = \pi_1 \cdots \pi_{m-1}$ and $\pi^R = \pi_{m+1} \cdots \pi_{n^*}$. We know that $S_2(\pi) = S_2(\pi^L)S_1(\pi^R)\pi_m = S_2(\pi^L)\pi^R n^*$. We wish to show that $S_2(\pi^L)\pi^R n^*$ avoids $\{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$ and is thus (k^*-1)-stack-sortable.

Suppose towards a contradiction that $S_2(\pi^L)\pi^R n^*$ contains some $\rho^* \in \{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$. Then either

1. $S_2(\pi^L)$ contains ρ^*

2. π^R contains ρ^*, or

3. $S_2(\pi^L)\pi^R$ contains ρ^*, where ρ^* uses numbers from both $S(\pi^L)$ and π^R.

Since π avoids $\{\rho \in \mathcal{S}_{k^*+2} | \rho_{k^*+2} = 1\}$, π^L avoids $\{\rho \in \mathcal{S}_{k^*+2} | \rho_{k^*+2} = 1\}$. Since π^L has length less than n^*, by the induction hypothesis, π^L must be k^*-stack-sortable. This means $S(\pi^L)$ is $(k^* - 1)$-stack-sortable, so $S(\pi^L)$ avoids $\{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$. Thus, case 1 is impossible.

If π^R contains some $\rho^* \in \{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$, then $n^* \pi^R$ contains some $\rho \in \mathcal{S}_{k^*+2}$ where $\rho_{k^*+2} = 1$, a contradiction. Thus, case 2 is impossible.

If $S_2(\pi^L)\pi^R$ contains some $\rho^* \in \{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$ then $\pi^L n^* \pi^R$ contains some $\rho \in \mathcal{S}_{k^*+2}$ where ρ_{k^*+2}=1, a contradiction. Thus, case 3 is impossible.

Since $S_2(\pi)$ avoids $\{\rho \in \mathcal{S}_{k^*+1} | \rho_{k^*+1} = 1\}$, by the induction hypothesis, $S_2(\pi)$ is $(k^* - 1)$-stack-sortable, and thus π is k^*-stack-sortable.

□

It turns out that permutations avoiding $\{\rho \in \mathcal{S}_{k+2} | \rho_{k+2} = 1\}$ have a simple enumeration. Avoiding these patterns means that 1 must be one of the first $(k + 1)$ elements of π, 2 must be one of the first $(k + 1)$ remaining elements, and so on. Ultimately, we achieve the exponential result below.

Corollary 9. *For $n \geq k$, there are $k!(k + 1)^{n-k}$ permutations of length n that are k-stack-sortable in a depth 2 stack. For $n < k$ there are $k!$ such permutations.*

4 2-Stack-Sortable Permutations in a Depth 3 Stack

We finally consider one additional case of k-stack-sorting in a stack of finite depth d that illustrates the complexity of the problem when we step away from the simplest cases of $k = 1$ and $d = 2$ detailed in the previous sections. Here we consider 2-stack-sorting with a stack of depth 3.

A case-by-case analysis yields the following characterization.

Lemma 10. *If $\pi \in \mathcal{S}_n$ avoids the patterns* 2341*,* $3\bar{5}241$*,* 25431*,* 52431*,* 54231*,* 463521*,* 465321*,* 563421*,* 564321*,* 645321*, and* 654321 *then π is 2-stack-sortable through a depth 3 stack.*

Proof. We prove this lemma by contrapositive. If π is not 2-stack-sortable, then by Theorem 4, $S_3(\pi)$ contains 231 or 4321. We now construct all possible ways for $S_3(\pi)$ to contain one of these two patterns:

Case 1: $S_3(\pi)$ contains an instance of 231 and the stack was not overflowed. By Theorem 2, π contains 2341 or $3\bar{5}241$.

Case 2: $S_3(\pi)$ contains an instance of 231 caused from stack overflow. Since the stack will overflow, at least three elements must occur before the 1 in decreasing order. Suppose that the 2 occurs after the 3 in π. Since the 2 is output before the 3, and the 231 pattern is caused by stack overflow, it must be the case that both 2 and 3 are in the stack simultaneously. If an element larger than 3 appears in input, the 231 pattern will not be caused by stack overflow, so once both 2 and 3 are in the stack, only elements smaller than 3 occur before the 1 in input. If there are only elements smaller than the 3 before 1 in the input, then the 1 will reach output before the 3 contradicting the formation of the 231 pattern in output. Therefore, the 2 must occur before the 3 in π and thus the 2 cannot be part of the descending pattern and we require a fifth element. There are three different possibilities: the 2 reaches the front of the input while the stack is empty (25431), or after a single item is in the stack (52431), or after the stack is full (54231).

Case 3: $S_3(\pi)$ contains an instance of 4321 and the stack was not overflowed. Since the stack was not overflowed, the elements must have popped from the stack prematurely because a bigger element appeared as the current input. Therefore, π must contain a 4321 pattern, and particularly placed elements propagate the 4321 to $S_3(\pi)$. However, placing these larger elements to pop the 3 before the 2 before the 1 means that there will be a 231 in $S_3(\pi)$; Case 1 handles this situation since stack overflow is not a concern at this point.

Case 4: $S_3(\pi)$ contains an instance of 4321 caused from stack overflow. As in Case 3, we must start with 4321 pattern already in π because the algorithm will always move larger elements to the right, not the left. Also, we need two more elements in decreasing order to overflow the stack, increasing our pattern length to 6. Therefore we begin with the subpattern 4321 and introduce 6 and 5 to create the troublesome patterns of length 6. We can have zero elements pop without a stack overflow (654321), or a single element $(465321, 645321)$, or two elements (463521). However, we can also use the 5 to play the role of the 4 in the 4321 pattern, and repeating the same technique, we find two more troublesome patterns $(564321, 563421)$. We cannot do the same with 6 representing 4, however, as we have no larger element that would force 6 out of the stack prematurely. □

As for the converse, we have the following result.

Lemma 11. *If $\pi \in \mathcal{S}_n$ contains one of the patterns 2341, $3\bar{5}241$, 25431, 52431, 54231, 465321, 645321, 654321, 563421, or 564321, then π is not 2-stack-sortable in a stack of depth 3.*

Proof. By Theorem 2, if π contains 2341 or $3\bar{5}241$, then π is not 2-stack-sortable in an infinite depth stack. Therefore, π is not 2-stack sortable in a stack of depth 3. Further, if $\pi \in \{25431, 52431, 54231\}$, then $S_3^2(\pi) = 21345$ and if $\pi \in \{465321, 645321, 654321, 563421, 564321\}$, then $S_3^2(\pi) = 213456$. Therefore π cannot is not 2-stack-sortable if it contains any of these patterns. □

We have now characterized those permutations sortable after 2 passes through a depth 3 stack in terms of 10 classical permutation patterns and 1 barred pattern. Magnusson and Ulfarsson [5] provide an alternate characterization of these permutations in terms of more compact decorated patterns. However, our characterization using classical and barred patterns allows us to use existing tools to enumerate these permutations exactly.

An *enumeration scheme* is an encoding for a system of recurrences that enumerates the members of a family of sets. Enumeration schemes were introduced by Zeilberger [14] and improved by Vatter [10] for the enumeration of permutations avoiding classical patterns. They were later extended by Pudwell [4] to enumerate permutations avoiding barred patterns. Using the `bVATTER` enumeration scheme algorithm in [4], we find a depth 5 enumeration scheme with 56 prefix patterns for permutations avoiding the set of patterns in Lemmas 10 and 11. The complete enumeration scheme is provided on the fifth author's website at http://faculty.valpo.edu/lpudwell/maple.html. This enumeration scheme has two nice properties in that (a) although this pattern set has a barred pattern, no stop points are required, and (b) every single reversibly deletable element is part of an interval of elements where new adjacent elements are forbidden by gap vectors. These two properties mean that the enumeration scheme produced by the `bVATTER` algorithm can be translated to a finitely labeled generating tree with 13 labels as shown below. The labels correspond to prefix permutations in the enumeration scheme after all relevant deletion maps have been applied.

- Root: (1)
- Rules:
  - $(1) \rightsquigarrow (12)(21)$

- $(12) \rightsquigarrow (12)(132)(231)$
- $(21) \rightsquigarrow (12)(312)(321)$
- $(132) \rightsquigarrow (12)(312)(321)(2431)$
- $(231) \rightsquigarrow (12)(2413)(3412)(3421)$
- $(312) \rightsquigarrow (12)(312)(321)(4231)$
- $(321) \rightsquigarrow (12)(312)(321)(4321)$
- $(2431) \rightsquigarrow (12)(312)(321)(2431)(2431)$
- $(2413) \rightsquigarrow (12)(312)(321)(2431)(2413)$
- $(3412) \rightsquigarrow (12)(2413)(3412)(3421)(3412)$
- $(3421) \rightsquigarrow (12)(2413)(3412)(3421)(3421)$
- $(4231) \rightsquigarrow (12)(312)(321)(4231)(4231)$
- $(4321) \rightsquigarrow (12)(312)(321)(4321)(4321)$

Using the transfer matrix method [8, Section 4.7] one can use these rules to compute the generating function for the permutations that are 2-stack-sortable in a depth 3 stack.

Theorem 12. *Let a_n be the number of permutations of length n that are 2-stack-sortable in a depth 3 stack. Then*

$$\sum_{n\geq 0} a_n x^n = \frac{2x^5 - 2x^4 - 3x^3 + 13x^2 - 7x + 1}{(x-1)(2x^4 - 12x^2 + 7x - 1)}.$$

5 Conclusion

We have now completely characterized and enumerated permutations that are k-stack-sortable through a depth d stack when $k = 1$, when $d = 2$, and when $k = 2$ and $d = 3$. In particular, the first two characterizations need only classical patterns and have particularly nice enumeration formulas. The enumeration in Section 4 utilizes algorithmic techniques to discover the 13-label generating tree needed to compute the necessary generating function and describes a sequence new to the literature.

The enumeration of permutations that are k-stack-sortable in a finite depth stack remains open for larger values of d and k. In particular, one would hope to find generating functions for specific d when $k = 2$ such that when one takes the limit as $d \to \infty$, we obtain a generating function for the values in Theorem 3. The enumeration scheme for $d = 3$ took 2 hours to compute on a 3.1 GHz laptop and used 56 prefix patterns of length 5 or less. Generalizing Lemma 10 is relatively straightforward: case 1 remains the same, case 3 is similar, but case 2 requires increasingly many patterns of length $d + 2$ and case 4 requires increasingly many patterns of length $2d$. Even for the case where $d = 4$, avoiding patterns of length 8 makes running the enumeration scheme algorithm prohibitive, and such an approach to computing the generating functions for $k = 2$ and $d \geq 4$ is beyond the practical time allowances of currently available computers when using the techniques of this paper.

When $k \geq 4$, the work of Magnusson and Ulfarsson suggest that eventually even decorated patterns become insufficient to characterize the sortable permutations compactly, so new techniques to approach both characterization and enumeration in these cases are desirable.

Certainly there are further patterns to be found in the spirit of the $s_{n,n-k}$ results of Section 2. To this end, we include brute force computational data for various choices of n, d, and k in the Appendix.

6 Appendix

In this appendix we provide the number of $\pi \in \mathcal{S}_n$ that are k-stack-sortable in a depth d stack, computed via brute force techniques. Note that if $d \geq n+k-1 \geq 2$, then the stack depth is irrelevant to whether $\pi \in \mathcal{S}_n$ is k-stack-sortable in a depth d stack and we obtain the number of permutations that are k-stack-sortable in an infinite depth stack. These values are shaded in gray; values where the stack depth limits the number of sortable permutations are in white.

$d\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	16	32	64	128	256	512	1024	2048
3	1	2	5	13	34	89	233	610	1597	4181	10946	28657
4	1	2	5	14	41	122	365	1094	3281	9842	29525	88574
5	1	2	5	14	42	131	417	1341	4334	14041	45542	147798
6	1	2	5	14	42	132	428	1416	4744	16016	54320	184736
7	1	2	5	14	42	132	429	1429	4846	16645	57686	201158
8	1	2	5	14	42	132	429	1430	4861	16778	58598	206516
9	1	2	5	14	42	132	429	1430	4862	16795	58766	207783
10	1	2	5	14	42	132	429	1430	4862	16796	58785	207990
11	1	2	5	14	42	132	429	1430	4862	16796	58786	208011
12	1	2	5	14	42	132	429	1430	4862	16796	58786	208012

Table 2: Number of 1-stack-sortable permutations in a finite depth stack

$d\backslash n$	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	2	6	18	54	162	486	1458	4374	13122
3	1	2	6	22	88	360	1480	6088	25036	102920
4	1	2	6	22	91	404	1865	8770	41564	197548
5	1	2	6	22	91	408	1933	9498	47757	243425
6	1	2	6	22	91	408	1938	9608	49162	257326
7	1	2	6	22	91	408	1938	9614	49328	259884
8	1	2	6	22	91	408	1938	9614	49335	260122
9	1	2	6	22	91	408	1938	9614	49335	260130
10	1	2	6	22	91	408	1938	9614	49335	260130

Table 3: Number of 2-stack-sortable permutations in a finite depth stack

$d\backslash n$	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	2	6	24	96	384	1536	6144	24576	98304
3	1	2	6	24	114	594	3244	17986	100126	557644
4	1	2	6	24	114	606	3474	20964	130714	830860
5	1	2	6	24	114	606	3494	21396	137044	907110
6	1	2	6	24	114	606	3494	21426	137859	921404
7	1	2	6	24	114	606	3494	21426	137901	922806
8	1	2	6	24	114	606	3494	21426	137901	922862
9	1	2	6	24	114	606	3494	21426	137901	922862
10	1	2	6	24	114	606	3494	21426	137901	922862

Table 4: Number of 3-stack-sortable permutations in a finite depth stack

$d\backslash n$	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	2	6	24	120	600	3000	15000	75000	375000
3	1	2	6	24	120	696	4416	29568	204040	1423856
4	1	2	6	24	120	696	4476	30984	226524	1724000
5	1	2	6	24	120	696	4476	31104	229650	1779082
6	1	2	6	24	120	696	4476	31104	229860	1785822
7	1	2	6	24	120	696	4476	31104	229860	1786158
8	1	2	6	24	120	696	4476	31104	229860	1786158
9	1	2	6	24	120	696	4476	31104	229860	1786158
10	1	2	6	24	120	696	4476	31104	229860	1786158

Table 5: Number of 4-stack-sortable permutations in a finite depth stack

$d\backslash n$	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	2	6	24	120	720	4320	25920	155520	933120
3	1	2	6	24	120	720	4920	36600	287760	2345336
4	1	2	6	24	120	720	4920	36960	297840	2534052
5	1	2	6	24	120	720	4920	36960	298680	2559612
6	1	2	6	24	120	720	4920	36960	298680	2561292
7	1	2	6	24	120	720	4920	36960	298680	2561292
8	1	2	6	24	120	720	4920	36960	298680	2561292
9	1	2	6	24	120	720	4920	36960	298680	2561292
10	1	2	6	24	120	720	4920	36960	298680	2561292

Table 6: Number of 5-stack-sortable permutations in a finite depth stack

References

[1] M. Dairyko, L. Pudwell, S. Tyner, and C. Wynn. Non-contiguous pattern avoidance in binary trees. *Electronic Journal of Combinatorics* **19** (2012), P22.

[2] S. Dulucq, S. Gire, and O. Guibert. A combinatorial proof of J. West's conjecture. *Discrete Math.* **187** (1998): 71-96.

[3] D.E. Knuth. The art of computer programming. Volume 3. Sorting and Searching. Addison- Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973.

[4] L. Pudwell. Enumeration schemes for permutations avoiding barred patterns. *Electronic Journal of Combinatorics* **17** (2010), R29.

[5] H. Magnusson, H. Ulfarsson. Algorithms for discovering and proving theorems about permutation patterns. arXiv:1211.7110, 2012.

[6] D. Rotem. Stack Sortable Permutations. *Discrete Math.* **33** (1981): 185–196.

[7] N. Sloane. The Encyclopedia of Integer Sequences. Available at http://oeis.org, 2013.

[8] R.P. Stanley. Enumerative combinatorics. Vol. 1, vol. 49 of Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 1997.

[9] H. Ulfarsson. Describing West-3-stack-sortable permutations with permutation patterns. *Séminaire Lotharingien de Combinatoire*, **67** (2012), Article B67d.

[10] V. Vatter. Enumeration schemes for restricted permutations. *Combinatorics, Probability, and Computing* **17** (2008): 137–159.

[11] J. West. Permutations with forbidden subsequences and stack-sortable permutations. Ph.D Thesis. The Massachusetts Institute of Technology, Cambridge, MA, 1990.

[12] J. West. Sorting Twice Through a Stack. *Theoretical Computer Science* **117.1-2** (1993): 303–313.

[13] D. Zeilberger. A proof of Julian West's conjecture that the number of two-stack sortable permutations of length n is $2(3n)!/((n+1)!(2n+1)!)$. *Discrete Math.* **102** (1992): 85-93.

[14] D. Zeilberger. Enumeration schemes, and more importantly, their automatic generation. *Annals of Combinatorics* **2** (1998), 185–195.