Permutations sorted by a finite and an infinite stack in series

Murray Elder¹ and Yoong Kuan Goh²

¹ University of Technology Sydney, Ultimo NSW 2007, Australia
murray.elder@uts.edu.au
² University of Technology Sydney, Ultimo NSW 2007, Australia
gohyoongkuan@gmail.com

Abstract. We prove that the set of permutations sorted by a stack of depth \( t \geq 3 \) and an infinite stack in series has infinite basis, by constructing an infinite antichain. This answers an open question on identifying the point at which, in a sorting process with two stacks in series, the basis changes from finite to infinite.

Keywords: patterns, string processing algorithms, pattern avoiding permutations, sorting with stacks.

1 Introduction

A permutation is an arrangement of an ordered set of elements. Two permutations with same relative ordering are said to be order isomorphic, for example, 132 and 275 are order isomorphic as they have relative ordering \( i < k < j \). A subpermutation of a permutation \( p_1 \ldots p_n \) is a word \( p_{i_1} \ldots p_{i_s} \) with \( 1 \leq i_1 < \cdots < i_s \leq n \). A permutation \( p \) contains \( q \) if it has a subpermutation that is order isomorphic to \( q \). For example, 512634 contains 231 since the subpermutation 563 is order isomorphic to 231. A permutation that does not contain \( q \) is said to avoid \( q \). Let \( S_n \) denote the set of permutations of \( \{1, \ldots, n\} \) and let \( S^\infty = \bigcup_{n \in \mathbb{N}} S_n \). The set of all permutations in \( S^\infty \) which avoid every permutation in \( \mathcal{B} \subseteq S^\infty \) is denoted \( \text{Av}(\mathcal{B}) \). A set of permutations is a pattern avoidance class if it equals \( \text{Av}(\mathcal{B}) \) for some \( \mathcal{B} \subseteq S^\infty \). A set \( \mathcal{B} = \{q_1, q_2, \ldots\} \subseteq S^\infty \) is an antichain if no \( q_i \) contains \( q_j \) for any \( i \neq j \). An antichain \( \mathcal{B} \) is a basis for a pattern avoidance class \( \mathcal{C} \) if \( \mathcal{C} = \text{Av}(\mathcal{B}) \).

Sorting mechanisms are natural sources of pattern avoidance classes, since (in general) if a permutation cannot be sorted then neither can any permutation containing it. Knuth characterised the set of permutations that can be sorted by a single pass through an infinite stack as the set of permutations that avoid 231 \[1\]. Since then many variants of the problem have been studied, for example \[1234\] \( 5 \) \( 6 \) \( 7 \) \( 8 \) \( 9 \) \( 10 \) \( 11 \) \( 12 \) \( 13 \) \( 14 \) \( 15 \) \( 16 \) \( 17 \) \( 18 \). The set of permutations sortable by a stack of depth 2 and an infinite stack in series has a basis of 20 permutations \[7\], while for two infinite stacks in series there is no finite basis \[12\]. For systems of a finite stack of depth 3 or more and infinite stack in series, it was not known whether the basis was finite or infinite.
Here we show that for depth 3 or more the basis is infinite. We identify an
infinite antichain belonging to the basis of the set of permutations sortable by a
stack of depth 3 and an infinite stack in series. A simple lemma then implies the
result for depth 4 or more. A computer search by the authors ([10]) yielded 8194
basis permutations of lengths up to 13 (see Table 1). Basis permutations are listed
at [https://github.com/gohyoongkuan/stackSorting-3](https://github.com/gohyoongkuan/stackSorting-3). The antichain used
to prove our theorem was found by examining this data and looking for patterns
that could be arbitrarily extended.

| Permutation length | Number of sortable permutations | Number of basis elements |
|--------------------|---------------------------------|-------------------------|
| 5                  | 120                             | 0                       |
| 6                  | 711                             | 9                       |
| 7                  | 4700                            | 83                      |
| 8                  | 33039                           | 169                     |
| 9                  | 239800                          | 345                     |
| 10                 | 1769019                         | 638                     |
| 11                 | 13160748                        | 1069                    |
| 12                 | 98371244                        | 1980                    |
| 13                 | 737463276                       | 3901                    |

2 Preliminaries

The notation \(\mathbb{N}\) denotes the non-negative integers \(\{0, 1, 2, \ldots\}\) and \(\mathbb{N}_+\) the positive
integers \(\{1, 2, \ldots\}\).

Let \(M_t\) denote the machine consisting of a stack, \(R\), of depth \(t \in \mathbb{N}_+\) and
infinite stack, \(L\), in series as in Fig. 1. A sorting process is the process of moving
entries of a permutation from right to left from the input to stack \(R\), then to
stack \(L\), then to the output, in some order. Each item must pass through both
stacks, and at all times stack \(R\) may contain no more than \(t\) items (so if at some
point stack \(R\) holds \(t\) items, the next input item cannot enter until an item is
moved from \(R\) to \(L\)).

A permutation \(\alpha = a_1a_2\ldots a_n\) is in \(S(t, \infty)\) if it can be sorted to 123\ldots\(n\)
using \(M_t\). For example, 243651 is in \(S(t, \infty)\) for \(t \geq 3\) since it can be sorted using
the following process: place 2, 4 into stack \(R\), move 4, 3, 2 across to stack \(L\), place
6, 5, 1 into stack \(R\), then output 1, 2, 3, 4, 5, 6. Note 243651 \(\not\in S(2, \infty)\) by [7].

The following lemmas will be used to prove our main result.

**Lemma 1.** Let \(\alpha = a_1a_2\ldots a_n\) in \(S(t, \infty)\) for \(t \in \mathbb{N}_+\). If \(i < j\) and \(a_i < a_j\) then
in any sorting process that sorts \(\alpha\), if both \(a_i\) and \(a_j\) appear together in stack \(L\)
then \(a_i\) must be above \(a_j\).

**Proof.** If \(a_j\) is above \(a_i\) in stack \(L\) then the permutation will fail to be sorted. □
Lemma 2. Let $\alpha = a_1a_2 \ldots a_n \in S(t, \infty)$ for $t \geq 3$ and suppose $1 \leq i < j < k \leq n$ with $a_i a_j a_k$ order-isomorphic to 132. Then in any sorting process that sorts $\alpha$, $a_i, a_j, a_k$ do not appear together in stack $R$.

Proof. If $a_i, a_j, a_k$ appear together in stack $R$, we must move $a_k$ then $a_j$ onto stack $L$ before we can move $a_i$, but this means $a_j, a_k$ violate Lemma 1. \hfill $\square$

Lemma 3. Let $\alpha = a_1a_2 \ldots a_n \in S(t, \infty)$ for $t \geq 3$ and $1 \leq i_1 < i_2 < \cdots < i_6 \leq n$ with $a_{i_1}a_{i_2} \ldots a_{i_6}$ order-isomorphic to 243651. Then in any sorting process that sorts $\alpha$, at some step of the process $a_{i_4}$ and $a_{i_5}$ appear together in stack $R$.

Proof. For simplicity let us write $a_{i_1} = 2, a_{i_2} = 4, a_{i_3} = 3, a_{i_4} = 6, a_{i_5} = 5, a_{i_6} = 1$.
Before 6 is input, 2, 3, 4 are in the two stacks in one of the following configurations:

1. 2, 4, 3 are all in stack $R$. In this case we violate Lemma 2.
2. two items are in stack $R$ and one is in stack $L$. In this case by Lemma 1 we cannot move 6 to stack $L$, so 6 must placed and kept in stack $R$. If $t = 3$ stack $R$ is now full, so 5 cannot move into the system, and if $t \geq 4$, when 5 is input we violate Lemma 2.
3. one item, say $a_i$, is in stack $R$ and two items are in stack $L$. In this case we cannot move 6, 5 into stack $L$ by Lemma 1 so they remain in stack $R$ on top of $a_i$, violating Lemma 2.
4. stack $R$ is empty. In this case, 2, 3, 4 must be placed in stack $L$ in order, else we violate Lemma 1. We cannot place 6, 5 into stack $L$ until it is empty, so they must both stay in stack $R$ until 4 is output.

In particular, the last case is the only possibility and in this case $a_{i_4}, a_{i_5}$ appear in stack $R$ together. \hfill $\square$

Lemma 4. Let $\alpha = a_1a_2 \ldots a_n \in S(t, \infty)$ for $t \geq 3$ and suppose $1 \leq i_1 < i_2 < \cdots < i_5 \leq n$ with $a_{i_1}a_{i_2} \ldots a_{i_6}$ order-isomorphic to 32514. Then, in any sorting process that sorts $\alpha$, if $a_{i_1}, a_{i_2}$ appear together in stack $R$, then at some step in the process $a_{i_3}, a_{i_4}$ appear together in stack $L$.\hfill $\square$
Proof. For simplicity let us write \(a_{i_1} = 3, a_{i_2} = 2, a_{i_3} = 5, a_{i_4} = 1, a_{i_5} = 4\). Figure 2 indicates the possible ways to sort these entries, and in the case that 2, 3 appear together in stack \(R\) we see that 4, 5 must appear in stack \(L\) together at some later point.

\[
\begin{align*}
\text{Fig. 2. Sorting 32514}
\end{align*}
\]

**Lemma 5.** Let \(\alpha = a_1 a_2 \ldots a_n \in S(t, \infty)\) for \(t \geq 3\) and suppose \(1 \leq i_1 < i_2 < \cdots < i_5 \leq n\) with \(a_{i_1} a_{i_2} \ldots a_{i_5}\) order-isomorphic to 32541. Then, in any sorting process that sorts \(\alpha\), if \(a_{i_1}, a_{i_2}\) appear together in stack \(L\), then at the step that \(a_{i_1}\) is output,

1. \(a_{i_3}, a_{i_4}\) are both in stack \(R\), and
2. if \(a_k\) is in stack \(L\) then \(k < i_2\).

Proof. For simplicity let us write \(a_{i_1} = 3, a_{i_2} = 2, a_{i_3} = 5, a_{i_4} = 4, a_{i_5} = 1,\) and \(\alpha = u_0 u_1 u_2 u_3 u_4 u_5\). Figure 3 indicates the possible ways to sort these entries. In the case that 2, 3 appear in stack \(R\) together, Lemma 4 ensures 2, 3 do
not appear together in stack $L$. In the other case, before 3 is moved into stack $L$, any tokens in stack $L$ come from $u_0u_1$. Thus when 3 is output the only tokens in stack $L$ will be $a_k$ with $k < i_2$. Lemma 1 ensures that 4, 5 are not placed on top of 3 in stack $L$, so that the step that 3 is output they sit together in stack $R$. □

Fig. 3. Sorting 32541

3 An infinite antichain

We use the following notation. If $\alpha = a_1 \ldots a_n$ is a permutation of $12 \ldots n$ and $m \in \mathbb{Z}$ then let $\alpha_m$ be the permutation obtained by adding $m$ to each entry of $\alpha$. For example $(1 2 3)_4 = 5 6 7$ and $13_{13} = 19$.

We construct a family of permutations $\mathcal{G} = \{ G_i \mid i \in \mathbb{N} \}$ as follows. Define

$$P = 2 4 3 7 6 1$$
$$x_j = (10 5 9)_{6j}$$
$$y_j = (13 12 8)_{6j}$$
$$S_i = (14 15 11)_{6i}$$
$$G_i = P x_0 y_0 x_1 y_1 \ldots x_i y_i S_i$$
The first three terms are

\[ G_0 = 2 4 3 7 6 1 (10 5 9) (13 12 8) 14 15 11, \]
\[ G_1 = 2 4 3 7 6 1 (10 5 9) (13 12 8) (16 11 15) (19 18 14) 20 21 17, \]
\[ G_2 = P (10 5 9) (13 12 8) (16 11 15) (19 18 14) (22 17 21)(25 24 20) 26 27 23. \]

A diagram of \( G_2 \) is shown in Figure 4 which shows the general pattern.

![Diagram of the permutation \( G_2 \)](image)

**Fig. 4.** Diagram of the permutation \( G_2 = 2 4 3 7 6 1 x_0 y_0 x_1 y_1 x_2 y_2 26 27 23 \)

We will prove that each \( G_i \) is an element of the basis of \( S(3, \infty) \) for all \( i \in \mathbb{N} \). Note that if we define \( x_{-1}, y_{-1} \) to be empty, \( G_{-1} = 243761895 \) is also an element of the basis. We noticed this and \( G_0 \) had a particular pattern which we could extend using \( x_j y_j \). However, we exclude \( G_{-1} \) from our antichain to make the proofs simpler.

**Proposition 6.** The permutation \( G_i \not\in S(3, \infty) \) for all \( i \in \mathbb{N} \).
Proof. Suppose for contradiction that \( G_i \) can be sorted by some sorting process. Since \( P \) is order isomorphic to 243651, by Lemma 3 in any sorting process 7, 6 appear together in stack \( R \). Next, 7 6 10 5 9 is order isomorphic to 32514 so by Lemma 4 since 7, 6 appear together in stack \( R \) we must have that 10, 9 appear together in stack \( L \) at some point in the process.

Now consider \( x_jy_j = (10 \ 5 \ 9 \ 13 \ 12 \ 8)_6 \), and assume that 10, 9, 6 both appear in stack \( L \) together. Since \( (10 \ 9 \ 13 \ 12 \ 8)_6 \) is order isomorphic to 32514 by Lemma 5, 13, 12 appear together in stack \( R \) and stay there until 10, 9 is output.

Next consider \( y_jx_{j+1} = (13 \ 12 \ 16 \ 11 \ 15)_6 \), and assume that 13, 12, 9 both appear in stack \( R \) together. Then since \( (13 \ 12 \ 16 \ 11 \ 15)_6 \) is order isomorphic to 32514 by Lemma 5, we have 16, 15, 10 appear together in stack \( L \). Note that 16, 15, 10 = 10, 9, 6, so putting the above observations together we see that for all \( 0 \leq j \leq i \) we have 10, 9, 6 both appear in stack \( L \) together and 13, 12, 9 appear together in stack \( R \) and stay there until 10, 9 is output.

Now we consider the suffix \( x_jy_jS_i = (10 \ 5 \ 9 \ 13 \ 12 \ 8 \ 14 \ 15 \ 11)_6 \)

where 10, 9, 6 are together in stack \( L \). Lemma 5 tells us not only that 13, 12, 9 appear together in stack \( R \) and stay there until 10 is output, but that anything sitting underneath 10 in stack \( L \) comes before 9 in \( G_i \), so in particular 14, 15 are not underneath 10. All possible processes to sort \( x_jy_jS_i \) are shown in Fig. 5. All possible sorting moves fail, which means \( G_i \) cannot be sorted.

The idea of the preceding proof can be summarised informally as follows. The prefix \( P \) forces 7, 6 to be together in stack \( R \), then Lemmas 4 and 5 alternately imply that the 10, 9, 6 terms of \( x_j \) must be in stack \( L \) and the 13, 12, 9 terms of \( y_j \) must be in stack \( R \). When we reach the suffix \( S_i \), the fact that certain entries are forced to be in a particular stack means we are unable to sort the final terms. We now show that if a single entry is removed from \( G_i \), we can choose to place the 10, 9, 6 terms in stack \( R \) and 13, 12, 9 terms in stack \( L \), which allows the suffix to be sorted.

**Lemma 7.** Let \( 0 \leq j \leq i \). If stack \( R \) contains one or both of 10, 9, 6 in ascending order, and \( y_j \ldots y_i S_i \) is to be input as in Fig. 5 then there is a sorting procedure to output all remaining entries in order.

**Proof.** For \( j < i \) move 13, 12 into stack \( L \), output 8, 9, 6, 10, move 16, 15, 14 into stack \( R \), output 11, 12, 5, 4, output 13, 12, 9 from stack \( L \) and input 15, 9 so that the configuration has the same form as Fig. 6 with \( j \) incremented by 1.

For \( j = i \) the remaining input is \((13 \ 12 \ 8 \ 14 \ 15 \ 11)_6 \). Put 13, 12 in stack \( L \) in order, output 8, 9, 6, 10, put 14, 15 in stack \( R \) and output 11, 12, 9, then 15, 9, move 15 into stack \( L \) and output 14 then 15.
Lemma 8. Let $0 \leq j \leq i$. If stack $L$ contains one or both of $12_6j, 13_6j$ in ascending order, and $x_{j+1}\ldots S_i$ (or just $S_i$ if $j = i$) is to be input as in Fig. 7, then there is a sorting procedure to output all remaining entries in order.

Proof. If $j < i$ move $10_6(j+1)$ into stack $R$, output $5_6(j+1), 12_6j, 13_6j$, move $9_6(j+1)$ to stack $R$ to reach the configuration in Fig. 9 which we can sort by Lemma 7. If $j = i$ then the remaining input is just $S_i = (14 15 11)_6$: move $14_6i, 15_6i$ to stack $R$, then output all entries.

If one of $12_6j, 13_6j$ is missing, use the same procedure ignoring the missing entry. \[\square\]

Proposition 9. Let $G'_i$ be a permutation obtained by removing a single entry from $G_i$. Then $G'_i \in S(3, \infty)$.

Proof. We give a deterministic procedure to sort $G'_i$. There are three cases depending on from where the entry is removed.

Term removed from $P$. Let $P'$ be the factor $P$ with one entry removed. We claim that there is a sorting sequence for $P'x_0$ which outputs the smallest six items in order and leaves $10, 9$ in stack $R$. To show this we simply consider all cases.
1. If 1 is removed, 2, 4, 3 can be output in order, then 7, 6 placed in stack $L$, 10 in stack $R$, then 5, 6, 7 output, and 9 placed on top of 10 in stack $R$.

2. If 2, 3, or 4 are removed, write $P' = ab761$ with $a, b \in \{2, 3, 4\}$. Place $a, b$ in stack $R$, move 7, 6 into stack $L$, output 1, then output $a, b$ in the correct order, then move 10 into stack $R$, output 5, 6, 7 and move 9 into stack $R$.

3. If 6 or 7 is removed, write $P' = 243a1$ with $a \in \{7, 6\}$. Place 4, 3, 2 in stack $L$ in order, move $a$ into stack $R$, output 1 then 2, 3, 4, then move $a$ into stack $L$, move 10 into stack $R$, output 5, $a$ and move 9 into stack $R$.

Thus after inputting $P'x_0$ we have the configuration shown in Fig. 6 with $j = 0$, which we can sort by Lemma 7.

Term removed from $x_s$, $0 \leq s \leq i$.

Input $P$ leaving 6, 7 in stack $R$, which brings us to the configuration in Fig. 8 with $j = 0$. Now assume we have input $P \ldots x_{j-1}y_{j-1}$ with $j \leq s$ (note the convention that $x_{-1}, y_{-1}$ are empty) and the configuration is as in Fig. 8.

If $j < s$ we can input $x_jy_j$ into the stacks to arrive at the same configuration with $j$ incremented by 1, as follows: move 10$_{6j}$ to stack $L$, output 5$_{6j}, 6_{6j} = 12_{6(j-1)}, 7_{6j} = 13_{6(j-1)}$, move 9$_{6j}$ to stack $L$, move 13$_{6j}, 12_{6j}$ to stack $R$, output 8$_{6j}, 9_{6j}, 10_{6j}$.

If $j = s$, we proceed as follows:
Fig. 8. Configuration after \( P \ldots x_{j-1}y_{j-1} \) is input

1. If \( 5_{6s} \) removed, output \( 6_{6s} = 12_{6(s-1)} \), \( 7_{6s} = 12_{6(s-1)} \), move \( 9_{6s}, 10_{6s} \) to stack \( R \), to reach the configuration in Fig. 6 with \( j = s \). From here the remaining entries can be sorted by Lemma 7.

2. If \( 10_{6s} \) is removed, output \( 5_{6s}, 6_{6s}, 7_{6s} \) and place \( 9_{6s} \) in stack \( R \), to reach the configuration in Fig. 6 with \( j = s \) and \( 10_{6s} \) missing. From here the remaining entries can be sorted Lemma 7.

3. If \( 9_{6s} \) is removed, move \( 6_{6s} \) to stack \( L \), move \( 10_{6s} \) on top of \( 7_{6s} \) in stack \( R \), output \( 5_{6s}, 6_{6s}, 13_{6s}, 12_{6s} \) into \( L \), then output \( 8_{6s}, 10_{6s} \). This gives the configuration in Fig. 7 with \( j = s \). From here the remaining entries can be sorted by Lemma 8.

Term removed from \( y_{s}, \) when \( 0 \leq s \leq i \) or \( S_{i} \). Input \( Px_{0} \) to reach the configuration in Fig. 9 with \( j = 0 \): move \( 2, 3, 4 \) into stack \( L \), \( 7, 6 \) to \( R \), output \( 1, 2, 3, 4 \), move \( 10 \) into \( L \), output \( 5, 6, 7 \) then move \( 9 \) into \( L \).

Fig. 9. Configuration after \( Px_{0}y_{0} \ldots x_{j} \) is input

Now suppose we have input \( Px_{0}y_{0} \ldots x_{j} \) to reach the configuration in Fig. 9. If no entry is removed from \( y_{j} \) and \( j < i \) then we can input \( y_{j}x_{j+1} \) to return to the configuration in Fig. 9 with \( j \) incremented by 1 as follows: move \( 13_{6j}, 12_{6j} \) to stack \( R \), output \( 8_{6j}, 9_{6j}, 10_{6j} \), move \( 10_{6(j+1)} \) to \( L \), output \( 5_{6(j+1)} = 11_{6j}, 12_{6j}, 13_{6j} \), then move \( 9_{6(j+1)} \) to stack \( L \).

If \( j = s \) (\( y_{s} \) is removed):
1. If $s_{6i}$ is removed, output $g_{6i}, 10_{6i}$. move $13_{6i}, 12_{6i}$ to stack $L$ to reach the configuration in Fig. 7 from which the remaining entries can be sorted by Lemma 8.

2. If $b \in \{13_{6i}, 12_{6i}\}$ is removed, place $b$ in stack $R$, output $8_{6i}, 9_{6i}, 10_{6i}$, move $b$ to stack $L$ to reach the configuration in Fig. 7 with one of $12_{6i}, 13_{6i}$ removed, from which the remaining entries can be sorted a by Lemma 8.

If $j = i$ and the entry is removed from $S_i$, sort the remaining entries as follows:

1. If $11_{6i}$ is removed, place $13_{6i}, 12_{6i}$ into stack $R$, output $8_{6i}, 9_{6i}, 10_{6i}, 11_{6i}$, then $12_{6i}, 13_{6i}, 14_{6i}, 15_{6i}$.

2. If $b \in \{14_{6i}, 15_{6i}\}$ is removed, place $13_{6i}, 12_{6i}$ into stack $R$, output $8_{6i}, 9_{6i}, 10_{6i}, 11_{6i}$, move $12_{6i}$ into stack $L$, place $b$ on top of $13_{6i}$ in stack $R$, output $11_{6i}$ then $12_{6i}$, move $b$ into stack $L$, output $13_{6i}$ then $b$.

\[ \square \]

**Theorem 10.** The set of permutations that can be sorted by a stack of depth 3 and an infinite stack in series has an infinite basis.

**Proof.** Proposition 6 shows that each $G_i$ cannot be sorted, and Proposition 9 shows that no $G_i$ can contain $G_j$ for $j \neq i$ as a subpermutation since any subpermutation of $G_i$ can be sorted. Thus $\mathcal{G} = \{G_i \mid i \in \mathbb{N}\}$ is an infinite antichain in the basis for $S(3, \infty)$.

\[ \square \]

4 From finite to infinitely based

Let $\mathcal{B}_t$ be the basis for $S(t, \infty)$ for $t \in \mathbb{N}_+$. Modifying Lemma 1 in [17] for the sorting case, we have the following:

**Lemma 11.** If $\sigma \in \mathcal{B}_t$ has length $n$ then either $\sigma$ or $(213)_n \sigma$ belongs to $\mathcal{B}_{t+1}$.

**Proof.** If $\sigma \notin S(t+1, \infty)$ then since $\sigma \in \mathcal{B}_t$, deleting any entry gives a permutation in $S(t, \infty) \subseteq S(t+1, \infty)$, so $\sigma \in \mathcal{B}_{t+1}$. Else $\sigma \in S(t+1, \infty)$. In any sorting process for $(213)_n \sigma$ the entries $1_n, 2_n, 3_n$ cannot appear together in stack $L$, so at least one entry must remain in stack $R$ which means we must sort $\sigma$ with stack $R$ of depth at most $t$, which is not possible, so $(213)_n \sigma$ cannot be sorted. If we remove an entry of the prefix then the two entries $a, b \in \{1_n, 2_n, 3_n\}$ can be placed in stack $L$ in order, leaving stack $R$ depth $t+1$ so the permutation can be sorted, and if an entry is removed from $\sigma$ then since $\sigma \in \mathcal{B}_t$ it can be sorted with $R$ having one space occupied.

\[ \square \]

**Theorem 12.** The set of permutations that can be sorted using a stack of depth $t \in \mathbb{N}_+$ and an infinite stack in series is finitely based if and only if $t \in \{1, 2\}$.

**Proof.** We have $|\mathcal{B}_1| = 1$ and $|\mathcal{B}_2| = 20$. Theorem 10 shows that $\mathcal{B}_3$ is infinite. Lemma 11 implies if $\mathcal{B}_t$ is infinite then so is $\mathcal{B}_{t+1}$.

A small modification of Propositions 6 and 9 shows that for $t \geq 4$ the set $\mathcal{G}_t = \{G_{i,t}\}$, where $G_{i,t} = P(x_{i00})\ldots(x_{iyi})(14 15 16 \ldots 12 11)_i$, is an explicit antichain in the basis of $S(t, \infty)$. Details can be seen in [10].
References

1. Albert, M., Atkinson, M., Linton, S.: Permutations generated by stacks and dequees. Ann. Comb. 14(1), 3–16 (2010), https://doi.org/10.1007/s00026-010-0042-9
2. Albert, M., Bousquet-Mélou, M.: Permutations sortable by two stacks in parallel and quarter plane walks. European J. Combin. 43, 131–164 (2015), https://doi.org/10.1016/j.ejc.2014.08.024
3. Atkinson, M.D., Livesey, M.J., Tulley, D.: Permutations generated by token passing in graphs. Theoret. Comput. Sci. 178(1-2), 103–118 (1997), http://dx.doi.org/10.1016/S0304-3975(96)00057-6
4. Atkinson, M.D., Murphy, M.M., Ruškuc, N.: Sorting with two ordered stacks in series. Theoret. Comput. Sci. 289(1), 205–223 (2002), http://dx.doi.org/10.1016/S0304-3975(01)00270-5
5. Bóna, M.: A survey of stack-sorting disciplines. Electron. J. Combin 9(2), A1 (2003)
6. Claesson, A., Dukes, M., Steingrímsson, E.: Permutations sortable by $n-4$ passes through a stack. Ann. Comb. 14(1), 45–51 (2010), https://doi.org/10.1007/s00026-010-0044-7
7. Elder, M.: Permutations generated by a stack of depth 2 and an infinite stack in series. Electron. J. Combin. 13(1), Research Paper #68 (2006), http://www.combinatorics.org/Volume_13/Abstracts/v13i1r68.html
8. Elder, M., Lee, G., Rechnitzer, A.: Permutations generated by a depth 2 stack and an infinite stack in series are algebraic. Electron. J. Combin. 22(2), Paper 2.16, 23 (2015)
9. Elvey-Price, A., Guttmann, A.J.: Permutations sortable by two stacks in series. Adv. in Appl. Math. 83, 81–96 (2017), https://doi.org/10.1016/j.aam.2016.09.003
10. Goh, Y.K.: Ph.D. thesis, University of Technology Sydney, in preparation.
11. Knuth, D.E.: The art of computer programming. Volume 3. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont. (1973), sorting and searching, Addison-Wesley Series in Computer Science and Information Processing
12. Murphy, M.M.: Restricted permutations, antichains, atomic classes, stack sorting. Ph.D. thesis, University of St Andrew (2002)
13. Pierrot, A., Rossin, D.: 2-stack pushall sortable permutations. CoRR abs/1303.4376 (2013), http://arxiv.org/abs/1303.4376
14. Pierrot, A., Rossin, D.: 2-stack sorting is polynomial. Theory Comput. Syst. 60(3), 552–579 (2017), https://doi.org/10.1007/s00224-016-9743-8
15. Smith, R.: Two stacks in series: a decreasing stack followed by an increasing stack. Ann. Comb. 18(2), 359–363 (2014), https://doi.org/10.1007/s00026-014-0227-8
16. Smith, R., Vatter, V.: The enumeration of permutations sortable by pop stacks in parallel. Inform. Process. Lett. 109(12), 626–629 (2009), https://doi.org/10.1016/j.ipl.2009.02.014
17. Tarjan, R.: Sorting using networks of queues and stacks. J. Assoc. Comput. Mach. 19, 341–346 (1972), https://doi.org/10.1145/321694.321704
18. West, J.: Sorting twice through a stack. Theoretical Computer Science 117(1), 303–313 (1993)