PU. M. A. Vol. **30** (2022), No.1, pp. 63 – 67

Preimages under a popqueue-sorting algorithm

Lapo Cioni
Dipartimento di Matematica e Informatica “U. Dini”
Università degli Studi di Firenze
Firenze, Italy
email: lapo.cioni@unifi.it

(Received: March 31, 2022, and in revised form May 15, 2022.)

Abstract. Following the footprints of what has been done with other sorting devices, we study a popqueue and define an optimal sorting algorithm, called `Cons`. Our results include a description of the set of all the preimages of a given permutation, an enumeration of the set of the preimages of permutations with some specific properties and, finally, the exact enumeration of permutations having 0, 1 and 2 preimages, respectively, with a characterization of permutations having 3 preimages.

Mathematics Subject Classification(2020). 05A05, 05A10, 05A15,

Keywords: sorting algorithm, permutation, popqueue, preimage.

1 Introduction

`Stacksort` is a classical and well-studied algorithm that attempts to sort an input permutation by (suitably) using a stack. It has been introduced and first investigated by Knuth [8] and West [9], and it is one of the main responsible for the great success of the notion of *pattern* for permutations. Among the many research topics connected with `Stacksort`, a very interesting one concerns the characterization and enumeration of preimages of the associated map, which is usually denoted with s (so that $s(\pi)$ is the permutation which is obtained after performing `Stacksort` on π). More specifically, given a permutation π, what is $s^{-1}(\pi)$? How many permutations does it contain? These questions have been investigated first by Bousquet-Melou [2], and more recently by Defant [6] and Defant, Engen and Miller [7].

The problem of studying the preimages of a sorting map has also been tackled for other sorting algorithms, such as `Queuesort` [4] and `Bubblesort` [3] [1].

In this paper we will consider another sorting device, namely a popqueue. A popqueue is a sorting device in which we can insert and extract elements, following some restrictions. These are the allowed operations, which can be seen in Figure 1:

- *enqueue*, insert the current element from the input into the popqueue, in the rightmost position;
- *pop*, remove all the elements currently in the popqueue, from left to right, sending them into the output;
- *bypass*, send the current element from the input directly into the output.

DOI: 10.2478/puma-2022-0010

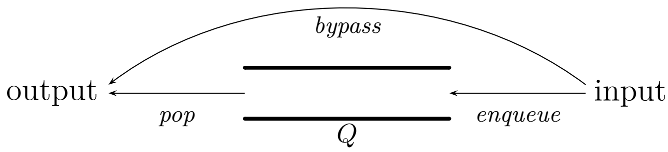

Figure 1: A popqueue.

These operations resemble those of a queue. Indeed, the sole difference is the fact that *pop* removes all the elements instead of removing only the leftmost one, which is the reason for the name "popqueue". This device is introduced and studied in [5]. Notice that, in the same fashion as with queues, we could instead have considered a sorting device consisting of two popqueues in pararallel, without bypass. Indeed, it is not difficult to see that such a device would have the same sorting power as the one we considered, since both are able to sort a permutation if and only if it avoids the patterns 321 and 2413.

To the best of our knowledge there is not a standard sorting algorithm for a popqueue. Below we describe the algorithm `Cons`, which takes its name from the fact that, during its execution, the content of the popqueue consists of consecutive values at all times. In the following description of `Cons`, $Front(Q)$ and $Back(Q)$ are the leftmost and the rightmost elements of the popqueue Q, respectively.

`Cons`
input: a permutation $\pi = \pi_1 \cdots \pi_n$
output: a permutation $\mathbf{C}(\pi)$
for $i = 1, \ldots, n$ do:

- if $\pi_i = Back(Q) + 1$, then *enqueue*;
- else, compare π_i and $Front(Q)$;
  - if $Front(Q) > \pi_i$, then *bypass*;
  - else, *pop* and *enqueue*.

Finally, *pop*.

`Cons` has some useful properties. First of all, it is an optimal sorting algorithm. This means that `Cons` sorts every permutation that can be sorted using a popqueue, which are all the permutations that avoid the patterns 321 and 2413 [5]. Also, recalling that a *left-to-right maximum* (*LTR maximum* for short) of a permutation $\pi = \pi_1 \ldots \pi_n$ is an element π_i such that $\pi_i > \pi_j$ for every $j < i$, we have that the elements that enter the popqueue during the sorting of π are precisely the LTR maxima of π.

In this work we will study the preimages of $\mathbf{C}$. Specifically, we will describe a procedure to find the preimages of an arbitrary permutation and we will also count them in some special cases. In order to do this, we need to give an alternative description of `Cons`, which focuses on the LTR maxima of the input permutation. Define $LTR(\pi)$ to be the set of LTR maxima of a permutation π.

2 The algorithm Cons

We now give a different description of `Cons`, which highlights how it behaves on the LTR maxima of a permutation. Let $\pi = \pi_1 \cdots \pi_n$. Mark π_1, and repeat the following steps until there are no marked elements:

- if there are no elements to the right of the (necessarily unique) block of consecutive marked elements, then unmark all marked elements;
- otherwise, compare the rightmost element μ of the block of consecutive marked elements with the element α to its right;
  - if $\mu > \alpha$, then swap α with the entire block of marked elements;
  - if $\mu = \alpha - 1$, then mark α;
  - if $\mu < \alpha - 1$, then mark α, and unmark all other elements of π.

Let's see how this procedure operates on a permutation, for example $\pi = 3241687$. The marked elements are indicated in **bold**.

$$\mathbf{3}241687 \to 2\mathbf{3}41687 \to 2\mathbf{34}1687 \to 21\mathbf{34}687 \to 2134\mathbf{6}87 \to 21346\mathbf{8}7 \to 213467\mathbf{8} \to 2134678$$

It is easy to see that the marked elements are always LTR maxima of the permutation. They are moved to the right until they reach the next LTR maximum; if this happens, they are glued together and continue moving to the right if and only if they are consecutive.

2.1 Preimages

We are interested in the preimages of $\mathbf{C}$. Specifically, given a permutation π, we want to describe the set $\mathbf{C}^{-1}(\pi)$ of the permutations whose output under `Cons` is π. We will start by stating some properties of the preimages and then we will give a procedure to find the preimages of any permutation π. Lastly, we will find some enumerative results about the number of permutations with exactly k preimages, for $k = 0, 1, 2, 3$.

Consider a permutation σ and set $\pi = \mathbf{C}(\sigma)$. Then the rightmost element of π is n. Indeed, if a permutation does not end with n, then it has no preimages under $\mathbf{C}$. Also, from the alternative description of `Cons`, we can see that $LTR(\sigma) \subseteq LTR(\pi)$, because the algorithm preserves the existing LTR maxima. Actually, there is a link between the subsets of $LTR(\pi)$ and the preimages of π, as we can describe the set $\mathbf{C}^{-1}(\pi)$ by looking at all the subsets of $LTR(\pi)$ and finding all the preimages (if any) with those LTR maxima. Notice that it is possible that there exists more than one preimage of π with the same set of LTR maxima, or there may not exist any at all. For example, 3421 and 3214 are both preimages of 2134 whose LTR maxima are 3 and 4. On the other hand, there are no preimages of 213 whose LTR maxima are both 2 and 3.

The following definition is the last tool that will allow us to describe all the preimages of a permutation having an arbitrary set of LTR maxima.

DEFINITION 2.1 Let $L = l_1 \cdots l_p$ and $A = a_1 \cdots a_r$ be two sequences of positive integers. Then we define the *mix* of L and A as the set $mix(L, A)$ of the shuffles of L and A whose first element is l_1.

For example, the mix of 245 and 13 is the set containing 24513, 24153, 24135, 21453, 21435, 21345.

Proposition 2.2 *Let π be a permutation ending with n. Let $B \subseteq LTR(\pi)$ such that $n \in B$.*

If there exist two elements $\mu, \lambda \in B$ such that $\lambda = \mu + 1$ and they are not consecutive in position in π, then there are no preimages of π whose set of LTR is B.

Otherwise, we can express π as $\pi = A_1L_1A_2L_2 \cdots A_kL_k$, where the blocks L_i are maximal sequences of consecutive elements of B. The blocks A_i contain the remaining elements of π, and may be empty. Then, all the preimages of π whose set of LTR maxima is B are those of the form $\pi = \rho_1\rho_2 \cdots \rho_k$, with $\rho_i \in mix(L_i, A_i)$ for every $i = 1, \dots, k$.

For example, for $\pi = 3245617$, if we select the subset $B = \{4, 5, 7\}$ we have that the corresponding preimages are 4532761, 4352761, 4325761. If we look at all the subsets of $LTR(\pi) = \{3, 4, 5, 6, 7\}$ we obtain the following preimages, which are all the preimages of π:

$\{7\}$: 7324561
$\{3, 7\}$: 3724561
$\{4, 7\}$: 4327561
$\{5, 7\}$: 5324761
$\{3, 5, 7\}$: 3524761
$\{4, 5, 7\}$: 4532761, 4352761, 4325761

3 Enumerative results

The correspondence between subsets of $LTR(\pi)$ and preimages of π would be simpler if there were no consecutive LTR maxima in π. Indeed, in that case there would be only one preimage for every subset. However this cannot happen, because a permutation has preimages if and only if it ends with its maximum n, therefore $n - 1$ is always a LTR maximum. Nonetheless we still have a simple case, described by the next proposition.

Proposition 3.1 *Let $\pi \in S_n$ be a permutation ending with n such that the only consecutive elements in $LTR(\pi)$ are $n - 1$ and n. Suppose that $n - 1$ and n are not consecutive in position in π, and let $k = |LTR(\pi)|$. Then $|\mathbf{C}^{-1}(\pi)| = 2^{k-2}$ and there is a bijection between $\mathbf{C}^{-1}(\pi)$ and the subsets of $LTR(\pi)$ containing n but not $n - 1$.*

Notice that the case described in the previous proposition gives a result that is analogous to the general result that counts the preimages of a permutation under `Bubblesort` [3]. In that case the number of preimages is 2^{k-1}, because the LTR maximum $n - 1$ does not give troubles and can be selected, and the way in which the preimages are constructed is the same. On the other hand, if there are consecutive elements in $LTR(\pi)$ other than $n-1$ and n, the analogy fails and there does not seem to be any link between the two situations.

Lastly, we provide results concerning permutations with a given number of preimages. Define $c_n^{(k)}$ as the number of permutations with length n which have exactly k preimages under `Cons`. We already noticed that a permutation has preimages if and only if it ends with its maximum, therefore $c_n^{(0)} = (n - 1)(n - 1)!$ for every $n \geq 1$, and $c_0^{(0)} = 0$.

The following propositions deal with permutations with exactly 1, 2 or 3 preimages.

PROPOSITION 3.2 *Let $\pi = \pi_1 \cdots \pi_n \in S_n$. Then π has exactly one preimage under $\mathbf{C}$ if and only if $\pi_n = n$ and $\pi_1 = n-1$, for every $n \geq 3$.*

Therefore $c_n^{(1)} = (n-2)!$ for every $n \geq 3$, and $c_0^{(1)} = 1$, $c_1^{(1)} = 1$, $c_2^{(1)} = 0$.

PROPOSITION 3.3 *Let $\pi = \pi_1 \cdots \pi_n \in S_n$. Then π has exactly two preimages under $\mathbf{C}$ if and only if $\pi_n = n$, $n-1 \neq \pi_1, \pi_{n-1}$ and $LTR(\pi) = \{\pi_1, n-1, n\}$, for every $n \geq 4$.*

Therefore $c_n^{(2)} = (n-2)! \sum_{j=1}^{n-3} \frac{1}{j}$ for every $n \geq 4$, and $c_0^{(2)} = 0$, $c_1^{(2)} = 0$, $c_2^{(2)} = 1$, $c_3^{(2)} = 0$.

PROPOSITION 3.4 *Let $\pi = \pi_1 \cdots \pi_n \in S_n$. Then π has exactly three preimages under $\mathbf{C}$ if and only if $\pi_n = n$, $LTR(\pi) = \{n-3, n-2, n-1, n\}$ and there are no LTR maxima which are positionally consecutive, for every $n \geq 4$.*

Looking at the previous results, we may wonder whether every number of preimages is allowed. That is, does there exist a permutation π such that $|\mathbf{C}^{-1}(\pi)| = k$, for every k?

For `Stacksort`, `Queuesort` and `Bubblesort` the answer is negative, so we could expect that also `Cons` may have some forbidden number. Surprisingly this is not the case, as a permutation of length $k-1$ with k preimages is $(k-3)\rho(k-2)(k-1)$, with $\rho \in S_{k-4}$, for $k \geq 5$. Since 3152647 has four preimages, every cardinality for the set of preimages is allowed.

References

[1] M. H. Albert, M. D. Atkinson, M. Bouvel, A. Claesson and M. Dukes, *On the inverse image of pattern classes under bubble sort*, J. Comb., 2 (2011) 231–243.

[2] M. Bousquet-Mélou, *Sorted and/or sortable permutations*, Discrete Math., 225 (2000) 25–50.

[3] M. Bouvel, L. Cioni and L. Ferrari, *Preimages under the bubblesort operator*, arXiv:2204.12936.

[4] L. Cioni and L. Ferrari, *Preimages under the Queuesort algorithm*, Discrete Math., 344 (2021) 112561.

[5] L. Cioni and L. Ferrari, *Sorting with a popqueue*, in preparation.

[6] C. Defant, *Postorder preimages*, Discrete Math. Theor. Comput. Sci., 19(1) (2017) #3.

[7] C. Defant, M. Engen and J. A. Miller, *Stack-sorting, set partitions, and Lassalle's sequence*, J. Combin. Theory Ser. A, 175 (2020) 105275.

[8] D. Knuth, *The Art of Computer Programming, Volume 1*, Boston: Addison-Wesley, 1968.

[9] J. West, *Permutations with forbidden subsequences and Stack sortable permutations*, PhD thesis, Massachusetts Institute of Technology, 1990.