Sorting Permutations with Fixed Pinnacle Set

Irena Rusu

LS2N, UMR6004, Université de Nantes, France
Irena.Rusu@univ-nantes.fr

Abstract

We give a positive answer to a question raised by Davis et al. (*Discrete Mathematics* 341, 2018), concerning permutations with the same $\pi_{i-1} < \pi_i > \pi_{i+1}$. The question is: given $\pi, \pi' \in S_n$ with the same pinnacle set S, is there a sequence of operations that transforms π into π' such that all the intermediate permutations have pinnacle set S? We introduce *balanced reversals*, defined as reversals that do not modify the pinnacle set of the permutation to which they are applied. Then we show that π may be sorted by balanced reversals (i.e. transformed into a standard permutation Id_S), implying that π may be transformed into π' using at most $4n - 2\min\{p, 3\}$ balanced reversals, where $p = |S| \geqslant 1$. In case $p = 0$, at most $2n - 1$ balanced reversals are needed.

Mathematics Subject Classifications: 05A05

1 Introduction

In a permutation $\pi = (\pi_1 \, \pi_2 \, \ldots \, \pi_n)$ from the symmetric group S_n, a *peak* is an index $i \neq 1, n$ such that $\pi_{i-1} < \pi_i > \pi_{i+1}$, whereas a *valley* is an index $i \neq 1, n$ such that $\pi_{i-1} > \pi_i < \pi_{i+1}$. *Descents* and *ascents* respectively identify indices i such that $\pi_i > \pi_{i+1}$ and $\pi_i < \pi_{i+1}$.

Many studies have been devoted to the combinatorics of peaks, especially to enumeration and counting problems [1, 2, 3, 6, 9, 11] (and many others). They identify strong and elegant relationships between peaks or descents in permutations, on the one hand, and Fibonacci numbers, Eulerian numbers, chains in Eulerian posets, etc. on the other hand.

In [5], Davis et al. revive the point of view considered in [4], and propose to identify peaks by their values rather than by their positions. They call a *pinnacle* any *element* π_i with $i \neq 1, n$ such that $\pi_{i-1} < \pi_i > \pi_{i+1}$, and show that considering pinnacles instead of peaks changes the combinatorial considerations behind counting and enumerating permutations with a given peak/pinnacle set. They characterize the sets of integers that may

be the pinnacle set of a permutation (so-called *admissible pinnacle sets*), count them, and propose bounds, involving the Stirling number, on the numbers of permutations with given pinnacle set.

They further ask several questions, one of which is considered in this paper:

Question 4.2 in [5]. For a given admissible pinnacle set S, is there a class of operations that one may apply to any $\pi \in S_n$ whose pinnacle set is S to obtain any other permutation $\pi' \in S_n$ with the same pinnacle set, in such a way that all the intermediate permutations have pinnacle set S?

This question is motivated by the search for similarities between pinnacles and peaks [5]. In this paper we give a positive answer to this question. More particularly, we identify a reduced set of reversals (the operation that reverses a block of a permutation) - called balanced reversals - which do not modify the set of pinnacles. Then we show that it is possible to transform any permutation with pinnacle set S into a canonical permutation of the same size with pinnacle set S by applying a sequence of at most $2n - \min\{p, 3\}$ balanced reversals. As the inverse transformation is always possible, this answers Question 4.2. above.

The paper is organized as follows. Section 2 contains the main definitions and notations. In Section 3 we identify balanced reversals and state the main results. Section 4 is devoted to the proof of our main theorem. This proof describes the algorithm allowing us to find the sequence of reversals transforming a given permutation into the canonical permutation. For the sake of completeness, we give in Section 5 the implementation details for an optimal running time of our algorithm. Section 6 is the conclusion.

2 Definitions and notations

Permutations π, π', π'' we use in the paper belong to the symmetric group S_n, for a given integer n. Elements $n + 1$ and $n + 2$ are artificially added at the beginning and respectively the end of each permutation, so that a permutation $\pi \in S_n$ is written as $\pi = (\pi_0 \, \pi_1 \, \pi_2 \, \ldots \, \pi_n \, \pi_{n+1})$ with $\pi_0 = n + 1$ and $\pi_{n+1} = n + 2$. For each $i > 0$, we define $Pred_\pi(\pi_i) = \pi_{i-1}$ and for each $i < n + 1$ we define $Next_\pi(\pi_i) = \pi_{i+1}$. The *block* of π with endpoints π_a and π_b (where $a \leqslant b$) is defined as $(\pi_a \, \pi_{a+1} \, \ldots \pi_{b-1} \, \pi_b)$.

A *pinnacle* is any *element* π_i with $i \neq 0, n + 1$ such that $\pi_{i-1} < \pi_i > \pi_{i+1}$. Similarly to pinnacles (whose indices are the peaks), we define the dells (whose indices are the valleys). A *dell* of π is any *element* π_i with $i \neq 0, n + 1$ such that $\pi_{i-1} > \pi_i < \pi_{i+1}$. The *shape* of the permutation π is the permutation $B_\pi = (y_0 \, v_1 \, y_1 \, v_2 \, y_2 \, \ldots, y_p \, v_{p+1} \, y_{p+1})$ where $v_1, \ldots, v_{p+1}$ are the dells of π, $y_1, \ldots, y_p$ are its pinnacles, whereas $y_0 = \pi_0 = n + 1$ and $y_{p+1} = \pi_{n+1} = n + 2$. The presence of the elements $n + 1$ and $n + 2$ at the beginning and the end of the permutation adds no pinnacle to the initial permutation, and ensures that dells v_1, v_{p+1} exist. Note that in a shape two consecutive dells are always separated by a pinnacle (and viceversa) since the first dell and its following element start a sequence of increasing elements, whereas the element preceding the second dell and the second dell itself end a sequence of decreasing elements. The elements belonging to both these

sequences is necessarily a pinnacle.

Moreover, let $A_\pi(v_i, y_i)$ with $1 \leqslant i \leqslant p+1$ be the set of elements in the block of π with endpoints v_i and y_i, which are neither a dell nor a pinnacle. Similarly let $D_\pi(y_i, v_{i+1})$ with $0 \leqslant i \leqslant p$ be the set of elements in the block of π with endpoints y_i and v_{i+1}, which are neither a dell nor a pinnacle. Sets $A_\pi()$ and $D_\pi()$ are respectively called *ascending* and *descending sets* of π. Note that y_0 and y_n belong respectively to the leftmost descending and the rightmost ascending set. The dells and pinnacles belong to no such set.

Example 1. Consider $\pi = (11\,8\,\underline{6}\,\overline{7}\,4\,3\,2\,\underline{1}\,5\,\overline{10}\,\underline{9}\,12)$ from S_{10} (thus $n = 10$) with elements 11 and 12 artificially added. Dells are underlined, pinnacles are overlined, $p = 2$. The shape is $B_\pi = (11\,\underline{6}\,\overline{7}\,\underline{1}\,\overline{10}\,\underline{9}\,12)$. The ascending sets are $A_\pi(6, 7) = \emptyset$, $A_\pi(1, 10) = \{5\}$ and $A_\pi(9, 12) = \{12\}$, whereas the descending sets are $D_\pi(11, 6) = \{8, 11\}$, $D_\pi(7, 1) = \{2, 3, 4\}$ and $D_\pi(10, 9) = \emptyset$.

We define a canonical permutation according to [5]. Given a set $S = \{s_1, s_2, \ldots, s_d\}$ and an integer $n > 2d$, the *canonical permutation $Id_S \in S_n$ with pinnacle set S* is the permutation built as follows: place the elements of S in increasing order on positions $2, 4, \ldots, 2d$ respectively; then place the elements in $\{1, 2, \ldots, n\} \setminus S$ in increasing order on positions $1, 3, \ldots, 2d-1, 2d+1, \ldots, n$. With our convention, elements $n+1$ and $n+2$ are added at the beginning and respectively the end of Id_S.

Definition 2. Let w_1, w_2 be two elements of π, such that $w_1 = \pi_a, w_2 = \pi_b$ and $0 < a \leqslant b < p+1$. The *reversal $\rho(w_1, w_2)$* is the operation that transforms

$$\pi = (\pi_0 \; \ldots \; \pi_{a-1} \; \underline{\pi_a \; \ldots \; \pi_b} \; \pi_{b+1} \; \ldots \; \pi_{n+1})$$

into

$$\pi' = (\pi_0 \; \ldots \; \pi_{a-1} \; \underline{\pi_b \; \pi_{b-1} \; \ldots \; \pi_{a+1} \; \pi_a} \; \pi_{b+1} \; \ldots \; \pi_{n+1}).$$

Notation: $\pi' = \pi \cdot \rho(w_1, w_2)$.

Example 3. With $S = \{7, 10\}$, the canonical permutation $Id_S \in S_{10}$ is $Id_S = (11\,\underline{1}\,\overline{7}\,\underline{2}\,\overline{10}\,\underline{3}\,4\,5\,6\;8\,9\,12)$, with shape $(11\,\underline{1}\,\overline{7}\,\underline{2}\,\overline{10}\,\underline{3}\,12)$. Recall that $\pi_0 = n+1 = 11$ and $\pi_{n+1} = n+2 = 12$ are added to each permutation in S_{10}. Then S has the same pinnacle set as π in Example 1, but not the same dells and thus not the same shape. Applying $\rho(1, 10)$ to Id_S yields the permutation $Id_S \cdot \rho(1, 10) = (11\,10\,\underline{2}\,\overline{7}\,\underline{1}\,3\,4\,5\,6\,8\,9\,12)$. It may be noticed that the resulting permutation has pinnacle set $\{7\}$, showing that reversals may modify the pinnacle set.

Definition 4. Let $\pi \in S_n$. A reversal $\rho(w_1, w_2)$ is a *balanced reversal* for π if π and $\pi \cdot \rho(w_1, w_2)$ have the same pinnacle set.

Balanced reversals are characterized in the next section. In order to identify appropriate balanced reversals when needed, we make use of cutpoints. Let i be an integer with $1 \leqslant i \leqslant p+1$ and z be an element of π not belonging to $A_\pi(v_i, y_i)$, such that $v_i < z < y_i$. The largest element e of $A_\pi(v_i, y_i) \cup \{v_i\}$ such that $e < z$ is called the *cutpoint* of z on

$A_\pi(v_i, y_i)$ and is denoted $cutA_\pi(z, v_i, y_i)$. The similar definition holds for $D_\pi(y_i, v_{i+1})$. Let i be an integer with $0 \leqslant i \leqslant p$ and z be an element of π not belonging to $D_\pi(y_i, v_{i+1})$, such that $v_{i+1} < z < y_i$. The largest element e of $D_\pi(y_i, v_{i+1}) \cup \{v_{i+1}\}$ such that $e < z$ is called the *cutpoint* of z on $D_\pi(y_i, v_{i+1})$ and is denoted $cutD_\pi(z, y_i, v_{i+1})$.

Example 5. Consider the permutation $\pi = (11\,7\,\underline{6}\,\overline{8}\,4\,3\,\underline{1}\,2\,5\,\overline{10}\,\underline{9}\,12)$ from S_{10}. Then we have $cutA_\pi(6, 1, 10) = cutA_\pi(9, 1, 10) = 5$ whereas $cutA_\pi(3, 1, 10) = 2$. Also, we have $cutD_\pi(6, 8, 1) = 4$ whereas $cutD_\pi(2, 8, 1) = 1$.

Finally, define the following problem:

BALANCED SORTING PROBLEM
Input: A permutation $\pi \in S_n$ with pinnacle set S.
Question: Is it possible to transform π into $Id_S \in S_n$ using only balanced reversals?

The difficulty in solving this problem has mainly two origins: first, one cannot perform any wished reversal since a reversal is not necessarily balanced (see Example 3); and second, given a set S of pinnacles and a permutation σ of the elements in S, it is possible that no permutation π of given size n and with pinnacle S exists that has the pinnacles in the order (from left to right) given by σ.

Example 6. Let set $S = \{3, 5, 7\}$. Then with $n = 7$ and $\sigma = (3\,5\,7)$ we may find the permutation $\pi = (8\,\underline{2}\,\overline{3}\,\underline{1}\,\overline{5}\,\underline{4}\,\overline{7}\,\underline{6}\,9)$, but with $n = 7$ and $\sigma = (3\,7\,5)$ there is no permutation from S_n with pinnacles in this order.

Therefore, the BALANCED SORTING PROBLEM is a question of feasibility in the first place. The optimal sorting is proposed as an open problem in the conclusion.

3 Main results

Let $\pi \in S_n$ be a permutation with pinnacle set S such that $|S| = p$. The main result of the paper is the following one.

Theorem 7. *There is a sequence R that solves the BALANCED SORTING PROBLEM on π using at most $2n - \min\{p, 3\}$ balanced reversals when $p \geqslant 1$, and at most $2n - 1$ reversals when $p = 0$.*

An answer to Question 4.2 is an immediate consequence of this theorem.

Corollary 8. *Let $\pi, \pi' \in S_n$ be two permutations with pinnacle set S such that $|S| = p$. Then, when $p \geqslant 1$ there is a sequence T of at most $4n - 2\min\{p, 3\}$ balanced reversals that transforms π into π', using only intermediate permutations with pinnacle set S. When $p = 0$, T contains at most $4n - 2$ balanced reversals.*
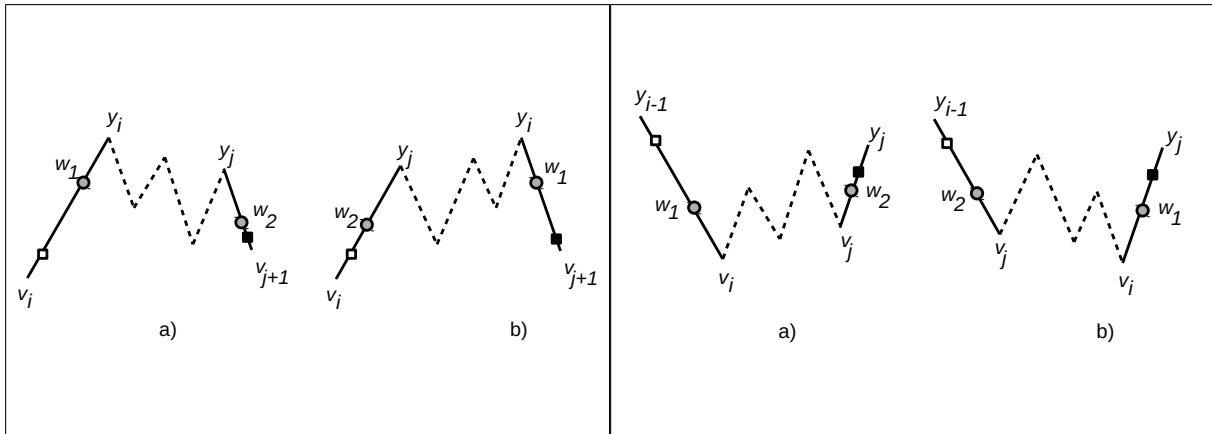
Figure 1: Intuitive description of types A.1 (left) and B.1 (right) where elements are placed on ascending and descending regions according to their values (high or low). A consequence is that neighboring elements on the permutation are not always at equal distance on the horizontal axis. Elements w_1 and w_2 are drawn as grey circles, $Pred_\pi(w_1)$ is drawn as a white square and $Next_\pi(w_2)$ is drawn as a black square. a) Permutation π. b) Result once the reversal $\rho(w_1, w_2)$ is applied.

Proof. Let R be the sequence of balanced reversals needed to sort π according to Theorem 7. Similarly, let $R' = (\rho(w_1, w_1'), \rho(w_2, w_2'), \ldots, \rho(w_q, w_q'))$ be the sequence of balanced reversals needed to sort π'. Let T be the sequence made of R followed by the sequence $\rho(w_q', w_q)$, $\rho(w_{q-1}', w_{q-1}), \ldots, \rho(w_1', w_1)$. Then T transforms π into Id_S and subsequently Id_S into π' using only balanced reversals. The definition of a balanced reversal guarantees that all the intermediate permutations have pinnacle set S. $\qquad\square$

Recall that, by Definition 2, in a reversal $\rho(w_1, w_2)$ the endpoints w_1 and w_2 are in this order from left to right on π and are distinct from y_0, y_{p+1}. Depending on the position of w_1 and w_2 in π, balanced reversals are of different types and imply different constraints, that need to be satisfied in order to guarantee that the reversal is balanced. Table 1 presents the different possible positions for w_1 and w_2, each defining a type. On the rightmost column are given the constraints that w_1, w_2 and their adjacent elements must fulfill in order to obtain a balanced reversal. For instance, reversal $\rho(w_1, w_2)$ of type A.1 is obtained when w_1 belongs to an ascending set of π and w_2 belongs to a descending set of π. One further requires that the following constraints be verified: when $Next_\pi(w_2) \neq v_{j+1}$ we must have $w_1 > Next_\pi(w_2)$; when $Pred_\pi(w_1) \neq v_i$ we must have $w_2 > Pred_\pi(w_1)$.

The standard cases A.1 and B.1 are shown in Figure 1. The other cases are obtained from A.1 or B.1 when w_1 or w_2 or both of them are a pinnacle or a dell. Cases denoted A.x are obtained from case A.1 only, cases B.x are obtained from B.1 only and cases C.x are obtained from both A.1 and B.1. Symmetrical cases are identified by an "s". We show below that these types form altogether the entire collection of balanced reversals.

Proposition 9. *Reversal $\rho(w_1, w_2)$ is balanced iff it belongs to the collection of types in Table 1.*

Type	Positions of w_1, w_2	Constraints
A.1	$w_1 \in A_\pi(v_i, y_i),$ $w_2 \in D_\pi(y_j, v_{j+1}), i \leqslant j$	if $Next_\pi(w_2) \neq v_{j+1}$ then $w_1 > Next_\pi(w_2)$ and if $Pred_\pi(w_1) \neq v_i$ then $w_2 > Pred_\pi(w_1)$
A.2	$w_1 = y_i, w_2 \in D_\pi(y_j, v_{j+1}), i \leqslant j$	$w_1 > Next_\pi(w_2)$
A.2s	$w_1 \in A_\pi(v_i, y_i), w_2 = y_j, i \leqslant j$	$w_2 > Pred_\pi(w_1)$
A.3	$w_1 = v_i, w_2 \in D_\pi(y_j, v_{j+1}), i \leqslant j$	if $Next_\pi(w_2) \neq v_{j+1}$ then $w_1 > Next_\pi(w_2)$ and if $Pred_\pi(w_1) = y_{i-1} \neq y_0$ then $w_2 < Pred_\pi(w_1)$
A.3s	$w_1 \in A_\pi(v_i, y_i), w_2 = v_{j+1}, i \leqslant j$	if $Pred_\pi(w_1) \neq v_i$ then $w_2 > Pred_\pi(w_1)$ and if $Next_\pi(w_2) = y_{j+1} \neq y_{p+1}$ then $w_1 < Next_\pi(w_2)$
B.1	$w_1 \in D_\pi(y_{i-1}, v_i), w_2 \in A(v_j, y_j), i \leqslant j$	$w_1 < Next_\pi(w_2)$ and $w_2 < Pred_\pi(w_1)$
B.2	$w_1 = y_i, w_2 \in A_\pi(v_j, y_j), i < j$	$Pred_\pi(w_1) = v_i, w_2 < Pred_\pi(w_1), Next_\pi(w_2) \neq y_j$ and $Next_\pi(w_2) < w_1$
B.2s	$w_1 \in D(y_{i-1}, v_i), w_2 = y_j, i \leqslant j$	$Pred_\pi(w_1) \neq y_{i-1}, Pred_\pi(w_1) < w_2, Next_\pi(w_2) = v_{j+1}$ and $w_1 < Next_\pi(w_2)$
B.3	$w_1 = v_i, w_2 \in A_\pi(v_j, y_j), i \leqslant j$	$w_2 < Pred_\pi(w_1)$ and if $Next_\pi(w_2) = y_j \neq y_{p+1}$ then $w_1 < Next_\pi(w_2)$
B.3s	$w_1 \in D_\pi(y_{i-1}, v_i), w_2 = v_j, i \leqslant j$	$w_1 < Next_\pi(w_2)$ and if $Pred_\pi(w_1) = y_{i-1} \neq y_0$ then $w_2 < Pred_\pi(w_1)$
C.1	$w_1 = v_i, w_2 = y_j, i \leqslant j$	$Pred_\pi(w_1) \neq y_{i-1}, w_2 > Pred_\pi(w_1)$ and $w_1 > Next_\pi(w_2)$
C.1s	$w_1 = y_i, w_2 = v_j, i \leqslant j$	$Next_\pi(w_2) \neq y_{j+1}, w_1 > Next_\pi(w_2)$ and $w_2 > Pred_\pi(w_1)$
C.2	$w_1 = v_i, w_2 = v_j, i < j$	if $Pred_\pi(w_1) = y_{i-1} \neq y_0$ then $w_2 < Pred_\pi(w_1)$ and if $Next_\pi(w_2) = y_{j+1} \neq y_{p+1}$ then $w_1 < Next_\pi(w_2)$
C.3	$w_1 = y_i, w_2 = y_j, i < j$	$w_1 > Next_\pi(w_2)$ and $w_2 > Pred_\pi(w_1)$

Table 1: Different types of balanced reversals. Each reversal is defined by constraints on w_1 and w_2, defining their places (middle column) and the relative orders required between some elements (rightmost column). Recall that w_1 and w_2 are in this order from left to right on π and are distinct from y_0, y_{p+1}. Then $Pred_\pi(w_1)$ and $Next_\pi(w_2)$ always exist.

Proof. "$\Rightarrow$": Several cases may appear.

If both w_1 and w_2 belong to ascending sets of the permutation, that is $w_1 \in A_\pi(v_i, y_i)$ and $w_2 \in A_\pi(v_j, y_j)$ with $i \leqslant j$, then when the reversal is performed w_2 or $Pred_\pi(w_1)$ becomes a new pinnacle, a contradiction. A similar reasoning holds when both w_1 and w_2 belong to descending sets of the permutation. So these cases cannot appear.

If one element among w_1 and w_2 belongs to an ascending set of the permutation, and the other one to a descending set of it, then it is easy to check that only the conditions in type A.1. or in type B.1. guarantee that no new pinnacle is added.

If exactly one element among w_1 and w_2 is a dell, then we necessarily have one of the types A.3, B.3, C.1 (or the symmetric ones) since any other condition creates a new pinnacle or removes an existing one.

If exactly one element among w_1 and w_2 is a pinnacle, then the other one is either a dell or belongs to an ascending or descending set of π. The former possibility necessarily leads to type C.1 (or the symmetric one). The latter possibility results into types A.2, A.2s, B.2 or B.2s.

If both w_1 and w_2 are pinnacles, or both are dells, then we must have the conditions in types C.2 or C.3 to preserve the pinnacle set.

$"\Leftarrow"$: This part only requires to check, for each type, that the pinnacle set is not modified under the indicated conditions. □

4 Proof of Theorem 7

We assume below that $p \geqslant 1$ and postpone the case $p = 0$ to Remark 19, at the end of the section.

In order to build the sequence R required in Theorem 7, we follow three steps:

1. Sort the p pinnacles of π in increasing order.

2. Place the wished dells (*i.e.* the dells of Id_S) as dells of π, in increasing order.

3. Move each element belonging to an ascending or descending set of π on the rightside of the last dell of π, in increasing order.

The result of these three steps is Id_S. Then R is the sequence of all the balanced reversals performed during these three steps in order to transform π into Id_S.

Remark 10. Note that in the subsequent, when permutation π is successively transformed using balanced reversals into some other permutation π', the elements of π' are identified both by their names in π', *i.e.* π'_1, π'_2 etc. and by their names in π, according to the needs. Once a given task is fulfilled by applying one or several balanced reversals, the resulting permutation is renamed as π, so that the following task begins with an initial permutation still denoted π.

4.1 Step 1: Sort the pinnacles

This is done by successively replacing the pinnacle y_k, for $k = 1, 2, \ldots, p-1$, by the k-th lowest pinnacle without modifying the set of pinnacles. Then in the resulting permutation the highest pinnacle is necessarily y_p. The other elements are not constrained at this step. Algorithm 1 presents the balanced rotations to be performed, as identified in this subsection.

Lemma 11. *There is a sequence of at most $p-1$ balanced reversals that transforms π with pinnacle S into π^* with pinnacle S such that y_1^* is the lowest pinnacle in π^*. Moreover, when $p \geqslant 3$, exactly one of the two following configurations occurs:*

(X) *the sequence contains exactly $p-1$ balanced reversals and y_p^* is the highest pinnacle in S.*

(Y) *the sequence contains at most $p-2$ balanced reversals.*

Proof. If y_1 is already the lowest pinnacle, then nothing is done. Assume now the lowest pinnacle is y_i with $i \neq 1$. Several cases are possible, that we present below, before giving the algorithm.

Case 1). $y_i < v_1$ *and for all* $h > i$*, we have* $y_h < v_1$*.*
    Then $v_{p+1} < y_p < v_1$ and the balanced reversal (type C.2) $\rho(v_1, v_{p+1})$ allows to obtain a permutation π' with $v_1' = v_{p+1} < v_1$. In π':

   a) if $y_i = y_1'$, then we are done.

   b) if $y_i \neq y_1'$ and $y_i < v_1'$, then we have that $y_{p+1}' = y_1 > v_1 > v_1'$ and we deduce that at least one pinnacle placed on the rightside of y_i is larger than v_1'. Then π' satisfies Case 2 below.

   c) if $y_i \neq y_1'$ and $y_i > v_1'$, then π' satisfies Case 3 below.

Case 2) $y_i < v_1$ *and there is* $h > i$ *such that* $y_h > v_1$*.*
    Then we assume w.l.o.g. that h is the minimum index with this property. Then $v_h < v_1$, otherwise we also have $y_{h-1} > v_h > v_1$ which contradicts the choice of h. Now, $\rho(v_1, v_h)$ is a balanced reversal (type C.2) since $Pred_\pi(v_1)$ is not a pinnacle and even if it may happen that $Next_\pi(v_h) = y_h$ we have $y_h > v_1$. Once $\rho(v_1, v_h)$ is performed, the new first dell is $v_1'' = v_h$ and is smaller than v_1 as proved above. Let us call π'' this new permutation, whose elements satisfy $y_t'' = y_i$ for some t, $y_{h-1}'' = y_1, y_h'' = y_h, y_1'' = y_{h-1}$. In π'':

   a) if $t = 1$, then we are done.

   b) if $t \neq 1$ and $y_i < v_1''$, then π'' satisfies Case 2 since $v_1'' = v_h < v_1 < y_1 = y_{h-1}''$, so there is at least one index as required in Case 2. The smaller such index, say g, satisfies $t < g < h$.

   c) if $t \neq 1$ and $y_i > v_1''$, then π'' is in Case 3 below.

    Condition $t < g < h$ in item b means that the recursivity we find here will end, as we show later (once Case 3 is presented).

Case 3) $y_i > v_1$*.*
    Let $e = cutA_\pi(y_i, v_1, y_1)$. Then $\rho(Next_\pi(e), y_i)$ is a balanced reversal (type A.2s if $Next_\pi(e) \neq y_1$ and type C.3 otherwise) since $y_i > e$ and $Next_\pi(e) > y_i > Next_\pi(y_i)$, both by the definition of the cutpoint e. This reversal places y_i as the leftmost pinnacle.

    The algorithm consists in applying Case 1 if necessary, then Case 2 as long as the current permutation requires it (in item b of Case 2) and finally Case 3 if needed. It is presented in Algorithm 1. In order to compute the number of balanced reversals performed in the worst case, we denote:

   • π the initial permutation

Algorithm 1 Permutation sorting by balanced reversals : Step 1

Input: A permutation $\pi \in S_n$ with pinnacle set S of cardinality p.
Output: The permutation π whose pinnacles have been placed in increasing order (Proposition 14)

1: $x \leftarrow \min\{y_1, \ldots, y_p\}$ // x is the lowest pinnacle
2: let $y_i = x$ // x has index i
3: **if** $(i \neq 1)$ and $y_i < v_1$ **then**
4:      $h \leftarrow \min\{h \mid h > i, y_h > v_1\} \cup \{0\}$ // $h = 0$ occurs when the first set is empty
5:      **if** $h = 0$ **then**
6:          $\pi \leftarrow \pi \cdot \rho(v_1, v_{p+1})$ // Case 1
7:      **end if**
8: **end if**
9: **while** x is not the leftmost pinnacle of π **do**
10:      let $y_i = x$ // x has index $i \neq 1$
11:      **if** $y_i < v_1$ **then**
12:          $h \leftarrow \min\{h \mid h > i, y_h > v_1\}$
13:          $\pi \leftarrow \pi \cdot \rho(v_1, v_h)$ //Case 2
14:      **else**
15:          $e \leftarrow cutA_\pi(y_i, v_1, y_1);\ \pi \leftarrow \pi \cdot \rho(Next_\pi(e), y_i)$ //Case 3
16:      **end if**
17: **end while**   // the lowest pinnacle is now placed in position y_1
18: **for** $k = 1$ to $p - 2$ **do**
19:      $x \leftarrow \min\{y_{k+1}, \ldots, y_p\}$; let $y_t = x$ // x is the lowest remaining pinnacle
20:      **if** $t \neq k + 1$ **then**
21:          $e \leftarrow cutA_\pi(y_t, v_{k+1}, y_{k+1});\ \pi \leftarrow \pi \cdot \rho(Next_\pi(e), y_t)$ //x is now in position y_{k+1}
22:      **end if**
23: **end for**
24: Return π

- π^0 the permutation obtained at the end of Case 1, whether it is applied or not (so that $\pi^0 = \pi$ if not).

- $\pi^1, \ldots, \pi^m$ the m successive permutations obtained using Case 2 ($m = 0$ if Case 2 is not applied).

- π^{m+1} the permutation obtained once Case 3 is applied, if it is applied.

As a consequence, if $m > 0$ then for $0 \leqslant q \leqslant m - 1$, permutation π^q is transformed into permutation π^{q+1} using the balanced reversal $\rho(v_1^q, v_{h^q}^q)$ of type C.2, as mentioned in Case 2 before. Here, h^q denotes the minimum index h computed in Case 2 for each π^q, *i.e.* $h^0 = h$ (see Case 2), $h^1 = g$ (see item b in Case 2), and so on. Indices $h^0, h^1 \ldots, h^{m-1}$ computed respectively in $\pi^0, \pi^1 \ldots, \pi^{m-1}$ satisfy (see again Case 2 item b where we show that $g < h$):

(i) $h^{m-1} < h^{m-2} < \cdots < h^1 < h^0$

(ii) $y_{h^q}^q, y_{h^{q-1}}^q, \ldots, y_{h^1}^q, y_{h^0}^q$ are pinnacles in the permutation π^q for each q with $0 \leqslant q \leqslant m-1$, in this order from left to right. Moreover, each permutation π^q inherits the pinnacles of the previous permutation π^{q-1}, that is, $y_{h^s}^q = y_{h^s}^{q-1}$ for all $0 \leqslant s \leqslant q-1$ (meaning that the pinnacles as well as their indices in the permutation are the same).

(iii) h^q and v_1^q satisfy the conditions of Case 2 item b in π^q for each q with $0 \leqslant q \leqslant m-1$.

(iv) permutation π^m obtained when Case 2 does not apply any longer contains all the pinnacles $y_{h^{m-1}}^m, y_{h^{m-2}}^m, \ldots, y_{h^1}^m, y_{h^0}^m$ built by the previous iterations, in this order from left to right.

The number of balanced reversals performed in this step strongly depends on the number m of reversals performed at worst in Case 2. By item (iv) above, $y_{h^m}^m, y_{h^m}^m, \ldots, y_{h^1}^m, y_{h^0}^m$ are m pinnacles in the permutation π^m obtained when Case 2 does not apply any longer, in this order from left to right. The number of such pinnacles (i.e. m) is upper bounded by $p-2$, since (1) $m \leqslant p$, and (2) when the m-th reversal is applied (to π^{m-1}), at least two pinnacles exist in the block to be reversed, the current leftmost pinnacle y_1^{m-1} and y_i. They are distinct, otherwise no reversal is applied. Thus $m \leqslant p-2$. Now:

- If $m = p-2$, then the pinnacles of π^{m-1} are necessarily, in this order from left to right, y_1^{m-1}, y_i ($= y_2^{m-1}$), $y_{h^{m-1}}^{m-1}$ ($= y_3^{m-1}$), $\ldots, y_{h^0}^{m-1}$ ($= y_p^{m-1}$). The last reversal due to Case 2, i.e. $\rho(v_1^{m-1}, v_{h^{m-1}}^{m-1})$, places y_i as the leftmost pinnacle, and thus we are done. Item a in Case 2 applies, and no other reversal is needed. Then the total number of reversals applied in Step 1 is $p-1$ when Case 1 applies and $p-2$ otherwise. In the latter case, configuration (Y) in the lemma occurs. The former case is fixed using property (P) below.

- If $m \leqslant p-3$, then we distinguish again several situations:

  − When $m \leqslant p-4$, the total number of reversals applied in Step 1 is $p-3$ when exactly one of Case 1 and Case 3 applies, and $p-2$ when both Case 1 and Case 3 apply. Configuration (Y) then occurs.

  − When $m = p-3$ and $h^0 \neq p$, then as above the pinnacles of π^{m-1} must be y_1^{m-1}, y_i ($= y_2^{m-1}$), $y_{h^{m-1}}^{m-1}$ ($= y_3^{m-1}$), $\ldots, y_{h^0}^{m-1}$($= y_{p-1}^{m-1}$) and y_p^{m-1}, where y_p^{m-1} is the rightmost pinnacle, that is never involved in the reversals. The last reversal due to Case 2, i.e. $\rho(v_1^{m-1}, v_{h^{m-1}}^{m-1})$, places y_i as the leftmost pinnacle, and thus we are done. Item a in Case 2 applies, and no other reversal is needed. Then the total number of reversals in step 1 is $p-2$ when Case 1 applies and $p-3$ otherwise, yielding configuration (Y) again.

  − When $m = p-3$ and $h^0 = p$, the total number of reversals applied in Step 1 is $p-2$ when at most one of Case 1 and Case 3 applies (configuration (Y) again), and $p-1$ when both Case 1 and Case 3 apply. The latter case is fixed using property (P) below.

We now finish the two unresolved cases, both of which occur when Case 1 applies and $y_p^m = y_{h^0}^m$ (recall that the pinnacles with indices $h^0, \ldots, h^{m-1}$ are inherited from one execution of Case 2 to the next one, by affirmation (ii) above).

(P) If Case 1 applies and in Case 2 we have $h_0 = p$, then y_p^m is the highest pinnacle in π^m.

Indeed, since Case 1 applies, in π we have $y_t < v_1$, for all $t \geqslant i$. Thus in π' (see Case 1) we have $y_u' < v_{p+1}' = v_1$ for all $u \leqslant s$, where $y_s' = y_i$. When π' is renamed as π^0, we have:

$$y_u^0 < v_{p+1}^0 = v_1 \text{ for all } u \leqslant s, \text{ where } y_s^0 = y_i. \tag{1}$$

Moreover, since Case 2 applies with $h^0 = p$, we have that:

$$y_p^0 > v_1^0 = v_{p+1} \tag{2}$$

$$y_r^0 < v_1^0, \text{ for all } r \text{ with } s \leqslant r < p. \tag{3}$$

By (1), $y_u^0 < v_{p+1}^0$ and by definition $v_{p+1}^0 < y_p^0$, thus $y_u^0 < y_p^0$ for all $u \leqslant s$, where $y_s^0 = y_i$. By (2) and (3), $y_r^0 < v_1^0 < y_p^0$ for all r with $s \leqslant r < p$. Thus y_p^0 is the highest pinnacle in π^0. Due to affirmation (ii) above, y_p^m is the highest pinnacle in π^m and property (P) is proved.

Now, property (P) applies in each of the two unresolved cases and yield configuration (X). Lemma 11 is proved. □

Once the lowest pinnacle is placed first, *i.e.* it is y_1, each of the other pinnacles is easily placed. The reasoning is by induction.

Lemma 12. *Assume that $y_1, y_2, \ldots, y_k$ are the k lowest pinnacles, with $k \geqslant 1$, and assume y_t with $t \neq k+1$ is the next lowest pinnacle. Then there is a balanced reversal allowing to replace y_{k+1} with y_t, which does not modify the pinnacles y_s, with $s \in \{1, \ldots, k, t+1, \ldots, p\}$.*

Proof. We notice that $y_{k+1} > y_t > y_k > v_{k+1}$. With $e = cutA_\pi(y_t, v_{k+1}, y_{k+1})$, the reversal $\rho(Next_\pi(e), y_t)$ is balanced (type A.2s if $Next_\pi(e) \neq y_{k+1}$ and type C.3 otherwise) and moves y_t at the sought place. □

Example 13. Consider $\pi = (20\ 16\ \underline{10}\ \overline{11}\ \underline{6}\ 17\ \overline{18}\ \overline{7}\ \overline{8}\ 1\ \overline{3}\ \underline{2}\ \overline{5}\ \underline{4}\ \overline{13}\ 12\ 9\ \underline{9}\ \overline{15}\ \underline{14}\ 19\ 21)$ Here, $n = 19$, $p = 7$, the dells are underlined and the pinnacles are overlined. Then $S = \{3, 5, 8, 11, 13, 15, 18\}$. The first and last elements are the bounds y_0 and y_{n+1} that are artificially added. Table 2 indicates the reversals needed to achieve step 1, according to Algorithm 1.

Lemmas 11 and 12 allow to deduce the following result.

Reversal	Permutation (once the reversal is performed)	Remarks
Initial	$\pi = (20\,16\,\underline{10}\,\overline{11}\,6\,17\,\overline{18}\,\overline{7}\,8\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{13}\,12\,9\,\overline{15}\,\underline{14}\,19\,21)$	Notations: dell: $\underline{10}$; pinnacle: $\overline{3}$
	Step 1	
$\rho(16,14)$	$\pi = (20\,\underline{14}\,\overline{15}\,9\,12\,\overline{13}\,4\,\overline{5}\,2\,\overline{3}\,1\,\overline{8}\,7\,\overline{18}\,17\,6\,\overline{11}\,\underline{10}\,16\,19\,21)$	Case 1 was applied. $x = y_4 = 3$.
$\rho(14,7)$	$\pi = (20\,\overline{7}\,\overline{8}\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{13}\,12\,9\,\overline{15}\,14\,\overline{18}\,17\,6\,\overline{11}\,\underline{10}\,16\,19\,21)$	Case 2 was applied.
$\rho(7,4)$	$\pi = (20\,4\,\overline{5}\,2\,\overline{3}\,1\,\overline{8}\,7\,\overline{13}\,12\,9\,\overline{15}\,14\,\overline{18}\,17\,6\,\overline{11}\,\underline{10}\,16\,19\,21)$	Case 2 here (and on the next line)
$\rho(4,1)$	$\pi = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,7\,\overline{13}\,12\,9\,\overline{15}\,14\,\overline{18}\,17\,6\,\overline{11}\,\underline{10}\,16\,19\,21)$	3 is now y_1; 5, 8 are also correct
$\rho(13,11)$	$\pi = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,7\,\overline{11}\,6\,17\,\overline{18}\,14\,\overline{15}\,9\,12\,\overline{13}\,\underline{10}\,16\,19\,21)$	11 is now y_4
$\rho(17,13)$	$\pi = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,7\,\overline{11}\,6\,\overline{13}\,12\,9\,\overline{15}\,14\,\overline{18}\,17\,\underline{10}\,16\,19\,21)$	13 is now y_5; 15, 18 are also correct
	Step 2	
$\rho(7,6)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,12\,9\,\overline{15}\,14\,\overline{18}\,17\,\underline{10}\,16\,19\,21) \doteq \pi$	$k=3, w=6, k+1=j-1=4$
$\rho(14,10)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,12\,9\,\overline{15}\,\underline{10}\,17\,\overline{18}\,14\,16\,19\,21) \doteq \pi$	$k=6, w=10, k+1=j-1=7$
$\rho(9,14)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,12\,14\,\overline{18}\,17\,\underline{10}\,\overline{15}\,9\,16\,19\,21)$	$k=7, w=12=Pred_\pi(9)$
$\rho(12,9)$	$\pi'' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,17\,\overline{18}\,14\,\underline{12}\,16\,19\,21) \doteq \pi$	
	Step 3	
$\rho(14,16)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,17\,\overline{18}\,16\,\underline{12}\,14\,19\,21)$	Item a), $u=16, e=14$
$\rho(12,14)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,17\,\overline{18}\,16\,14\,\underline{12}\,19\,21) \doteq \pi$	
$\rho(17,18)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,\overline{18}\,17\,16\,14\,\underline{12}\,19\,21)$	Item b), $i=7, u=17, e=16$
$\rho(18,18)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,\overline{18}\,17\,16\,14\,\underline{12}\,19\,21) \doteq \pi$	trivial
$\rho(17,12)$	$\pi' = (20\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{8}\,6\,\overline{11}\,7\,\overline{13}\,9\,\overline{15}\,\underline{10}\,\overline{18}\,\underline{12}\,14\,16\,17\,19\,21) \doteq \pi$	Item c)
	$\pi = Id_S$	Item d) does nothing here

Table 2: Execution of Steps 1, 2 and 3 on the permutation $\pi = (20\,16\,\underline{10}\,\overline{11}\,\underline{6}\,17\,\overline{18}\,\overline{7}\,8\,1\,\overline{3}\,2\,\overline{5}\,4\,\overline{13}\,12\,9\,\overline{15}\,\underline{14}\,19\,21)$. Notation $\pi' = (\ldots) \doteq \pi$ means that once π' is computed according to the algorithm, the algorithm does not compute π'' and thus π' is renamed π. Notation $\pi'' = (\ldots) \doteq \pi$ means that both π' and π'' have been computed, and π'' is renamed π.

Proposition 14. *There is a sequence R_1 of at most one reversal (when $p = 2$), and at most $2p - 4$ reversals (when $p \geqslant 3$) allowing to order the pinnacles of π in increasing order.*

Proof. When $p = 2$, Lemma 11 guarantees that at most $p - 1(= 1)$ reversals are needed.

For $p \geqslant 3$, in configuration (X) from Lemma 11 the leftmost pinnacle is already the highest one, so that in Step 18 of Algorithm 1 the last execution (for $k = p - 2$) will find the pinnacle y_{p-1} already on its place (since $y_1, \ldots, y_{p-2}$ and y_p are already correctly placed). Therefore, only $p - 3$ applications of Lemma 12 are required in this case. The total number of reversals is then $(p - 1) + (p - 3) = 2p - 4$.

For $p \geqslant 3$, in configuration (Y) from Lemma 11, we apply Lemma 12 for each k in $\{1, 2, \ldots, p - 2\}$ (see Algorithm 1). The number of reversals is then at most $(p - 2) + (p - 2) = 2p - 4$. $\qquad\square$

4.2 Step 2: Place the wished dells

Now we replace $v_1, v_2, \ldots, v_{p+1}$ respectively with the lowest, the second lowest etc. element which is not a pinnacle, in order to have in π the same dells as in Id_S. To this end, we need the following technical lemmas.

Algorithm 2 ApplyLemma15

Input: A permutation $\pi \in S_n$, pinnacles v_i, v_q satisfying the hypothesis of Lemma 15.
Output: The permutation π'' obtained according to Lemma 15.

1: $u = Pred_\pi(v_i)$
2: **if** $u > v_q$ **then**
3:     $e \leftarrow cutA_\pi(u, v_q, y_q)$; $\pi' \leftarrow \pi \cdot \rho(u, e)$; $\pi'' \leftarrow \pi' \cdot \rho(e, v_i)$;   //case i)
4: **else**
5:     $\pi' \leftarrow \pi \cdot \rho(v_i, v_q)$; $\pi'' \leftarrow \pi' \cdot \rho(u, v_i)$ //case ii)
6: **end if**
7: Return π''

Lemma 15. *Assume $v_i \leqslant v_q$, with $i \leqslant q$, such that $Pred_\pi(v_i)$ is not a pinnacle and satisfies $y_0 \neq Pred_\pi(v_i) < y_q$. Then there exist two balanced reversals transforming π into π'' such that the **only** differences between π and π'' are the following ones:*

   i) if $Pred_\pi(v_i) > v_q$, then $Pred_\pi(v_i)$ is moved immediately after $cutA_\pi(Pred_\pi(v_i), v_q, y_q)$, so that $Pred_\pi(v_i) \in A_{\pi''}(v_q, y_q)$.

   ii) if $Pred_\pi(v_i) < v_q$, $Pred_\pi(v_i)$ is moved immediately after v_q and becomes v_q''.

Proof. Let $u = Pred_\pi(v_i)$. See Algorithm 2.

i) If $u > v_q$ then let $e = cutA_\pi(u, v_q, y_q)$. Then $\rho(u, e)$ is a balanced reversal (type B.1 if $e \neq v_q$ or type B.3s otherwise), since $u < Next_\pi(e)$ and $e < u < Pred_\pi(u)$, both by the definition of the cutpoint e. When applied, this reversal yields a permutation π' where $v_i' = v_q, e \in D_{\pi'}(y_{i-1}', v_i'), Pred_{\pi'}(e) = Pred_\pi(u), v_q' = v_i$ and $Next_{\pi'}(v_i) = u$. Then $\rho(e, v_i)$ is a balanced reversal (type B.3s if $e \neq v_q$ and type C.2 otherwise) in π' for we have $e < Next_{\pi'}(v_i) = u$ by the definition of the cutpoint e, and $v_i < u < Pred_\pi(u) = Pred_{\pi'}(e)$. The resulting permutation π'' satisfies the conditions in the lemma.

ii) If $u < v_q$ then $\rho(v_i, v_q)$ is a balanced reversal (type C.2) since $Pred_\pi(v_i) \neq y_{i-1}$, and we have $v_i < u < v_q < Next_\pi(v_q)$ whether $Next_\pi(v_q) = y_q$ or not. In the permutation π' resulting once $\rho(v_i, v_q)$ is applied, $v_i' = u$ (since $u < v_q$), $Next_{\pi'}(v_i') = v_q, v_q' = v_i, Next_{\pi'}(v_i) = Next_\pi(v_q)$. Then $\rho(u, v_i)$ is a balanced reversal (type C.2) in π'. To see this, we need to show that $v_i < Pred_\pi(u)$ which is true since $Pred_\pi(u) > u > v_i$ in π, and that $u < Next_{\pi'}(v_i)$ which is also true since $Next_{\pi'}(v_i) = Next_\pi(v_q) > v_q > u$. The permutation π'' obtained after the execution of the reversal $\rho(u, v_i)$ satisfies the conditions in the lemma. $\qquad\square$

Lemma 16. *Assume $v_i \leqslant v_q$, with $i \leqslant q$, such that $Next_\pi(v_i)$ is not a pinnacle and satisfies and $Next_\pi(v_i) < y_{q-1}$. Then there exist two balanced reversals transforming π into π'' such that the **only** differences between π and π'' are the following ones:*

   i) if $Next_\pi(v_i) > v_q$, then $Next_\pi(v_i)$ is moved immediately before $cutD_\pi(Next_\pi(v_i), y_{q-1}, v_q)$, so that $Next_\pi(v_i) \in D_{\pi''}(y_{q-1}, v_q)$,

---

Algorithm 3 ApplyLemma16

Input: A permutation $\pi \in S_n$, pinnacles v_i, v_q satisfying the hypothesis of Lemma 16.
Output: The permutation π'' obtained according to Lemma 16.

1: $u = Next_\pi(v_i)$
2: **if** $u > v_q$ **then**
3: $\quad e \leftarrow cutD_\pi(u, y_{q-1}, v_q)$ //case i)
4: $\quad \pi' \leftarrow \pi \cdot \rho(u, Pred_\pi(e)); \pi'' \leftarrow \pi' \cdot \rho(Pred_\pi(e), Next_\pi(u))$
5: **else**
6: $\quad \pi' \leftarrow \pi \cdot \rho(u, Pred_\pi(v_q))$ //case ii)
7: $\quad \pi'' \leftarrow \pi' \cdot \rho(Pred_\pi(v_q), Next_\pi(u))$
8: **end if**
9: Return π''

---

   ii) if $Next_\pi(v_i) < v_q$, $Pred_\pi(v_i)$ is moved immediately before v_q and becomes v_q''.

Proof. Let $u = Next_\pi(v_i)$. See Algorithm 3.

   i) If $u > v_q$ then let $e = cutD_\pi(u, y_{q-1}, v_q)$ et $f = Pred_\pi(e)$. Then $\rho(u, f)$ is a balanced reversal (type A.1 if $f \neq y_{q-1}$, type A.2s otherwise), since $u > e$ and $f > u > Pred_\pi(u)$, both by the definition of a cutpoint and whether $f = y_{q-1}$ or not. When applied, this reversal yields a permutation π' where $v_i' = v_i$, $Next_\pi(v_i') = f$ with $f \in A_{\pi'}(v_i, y_{q-1})$ or $f = y_{q-1}$, as well as $v_q' = v_q$, $y_{q-1}' = y_i$, and $u \in D_{\pi'}(y_i, v_q)$ with $Next_{\pi'}(u) = e$ and $Pred_{\pi'}(u) = Next_\pi(u)$. Let $t = Next_\pi(u)$. Then $t = Pred_{\pi'}(u)$ and $\rho(f, t)$ is a balanced reversal in π' (type A.1 if $t \neq y_i$ and $f \neq y_{q-1}$, type A.2 or A.2s if exactly one equality holds and type C.3. otherwise). Indeed, in all types but A.2 we need to show that $t > Pred_{\pi'}(f)$ and this is true since $Pred_{\pi'}(f) = v_i' = v_i < u < t$, because v_i, u, t occur in this order on $A_\pi(v_i, y_i)$. Moreover, types A.1, A.2 and C.3 require that $f > Next_{\pi'}(t)$, which is true since $Next_{\pi'}(t) = u$ and $u < f$ by the definition of the cutpoint e. The permutation π'' resulting once $\rho(f, t)$ is performed satisfies the conditions in the lemma.

   ii) If $u < v_q$ then $\rho(u, Pred_\pi(v_q))$ is a balanced reversal (type A.1 if $Pred_\pi(v_q) \neq y_{q-1}$, type A.2s otherwise). Indeed, with the notation $s = Pred_\pi(v_q)$, in case that $s \neq y_{q-1}$ the reversal is of type A.1. and we have $Next_\pi(s) = v_q$ and $Pred_\pi(u) = v_i$, so that the conditions in type A.1 are trivially verified. If $s = y_{q-1}$ we have to check for type A.2s that $s > Pred_\pi(u)$, which is true as $s > v_q > u > v_i = Pred_\pi(u)$. In the permutation π' resulting once $\rho(u, s)$ is applied, $v_i' = v_i$, $Next_\pi(v_i') = s \in A_{\pi'}(v_i, y_{q-1})$ since $s > v_q > u > v_i$, $v_q' = u$, deduced because $u < v_q$, and $Next_{\pi'}(u) = v_q$. With $t = Next_\pi(u)$, we also have that $t = Pred_{\pi'}(u)$. Then, in π', $\rho(s, t)$ is a balanced reversal (type A.1 if $t \neq y_i$ and $s \neq y_{q-1}$, type A.2 or A.2s if exactly one equality occurs, resp. type C.3 if both equalities occur). Type A.1 is trivially verified, types A.2 and A.2s are guaranteed by $s > v_q > u$ respectively $t = Next_\pi(u) > u = Next_\pi(v_i) > v_i$, whereas type C.3 is guaranteed by the latter two conditions together. Once this reversal is applied, the resulting permutation π'' satisfies the lemma. $\qquad\square$

   We are now able to place the dells in Id_S as dells of π, in increasing order from left to

right according to the method described in Proposition 17 below and its proof. Algorithm 4 presents the approach. The continued example in Table 2 illustrates it.

Proposition 17. *Let $\pi \in S_n$ be a permutation with pinnacles $y_1 < y_2 < \ldots < y_p$. There is a sequence R_2 of at most $2p+2$ balanced reversals that places the dells in Id_S as dells of π, in increasing order from left to right, without modifying the pinnacles of π (nor their order).*

Proof. Assume the lowest $k < p + 1$ dells ($k = 0$ is admitted here) are correctly placed as $v_1, v_2, \ldots, v_k$ respectively, and let w be the next lowest element in π which is not a pinnacle. Then w must replace v_{k+1}. We have that $w < y_k$ since $w < v_{k+1} < y_k$.

Then either w is adjacent to a dell among $v_1, v_2, \ldots, v_k$, or w is itself a dell. In all the other cases, a smaller element would be found, contradicting the choice of w.

Case 1. w is adjacent to a dell

Let $w = Pred_\pi(v_j)$ or $w = Next_\pi(v_j)$ for some j with $1 \leqslant j \leqslant k$. Then we use Lemma 15, respectively Lemma 16 with $i = j$ and $q = k + 1$. We have that $v_j < v_{k+1}$ by the minimality of the elements $v_1, v_2, \ldots, v_k$. We also have $w < y_k$ as proved above, and thus $w < y_{k+1}$ by the increasing order of the pinnacles. So the hypothesis of the appropriate Lemma is satisfied. As $w < v_{k+1}$ by the minimality of w, item ii) of the lemma holds. Consequently, after two balanced reversals, π'' is the same as π except that w has been removed from its place and has been placed before or after v_{k+1} (depending on which lemma is applied), thus becoming the dell v''_{k+1}. Then we are done in this case.

Case 2. w is a dell

If w is already a dell, let $w = v_j$ with $k + 1 < j \leqslant p + 1$. Then $w < v_{k+1}$ by the minimality of w and $v_{k+1} < y_j$ since $v_{k+1} < y_{k+1} \leqslant y_j$. Let $e = cutA_\pi(v_{k+1}, v_j, y_j)$ and, if it exists, $f = Next_\pi(e)$. Then $\rho(v_{k+1}, e)$ is a balanced reversal (type B.3 if $e \neq v_j$ and type C.2 otherwise) since $e \neq y_j$ (v_{k+1} is an intermediate value among them), $e < v_{k+1} < Pred_\pi(v_{k+1})$ and $v_{k+1} < f$ by the definition of the cutpoint e. In the permutation π' resulting once $\rho(v_{k+1}, e)$ is applied, $v'_{k+1} = v_j, y'_{k+1} = y_{j-1}, y'_{j-1} = y_{k+1}, v'_j = v_{k+1}$ and $y'_j = y_j$. If $k + 1 = j - 1$, then the order of the pinnacles does not change since the reversed block contains a unique pinnacle. In this case we are done. Otherwise, due to $k + 1 < j - 1$ we deduce that $y_{k+1} < y_{j-1}$ and thus $v_{k+1} < y_{k+1} < y_{j-1}$. The cutpoint defined as $e' = cutA_{\pi'}(y_{k+1}, v_j, y_{j-1})$ satisfies then the condition $e' \in \{v_j\} \cup A_{\pi'}(v_j, y_{j-1})$. As a consequence, with $f' = Next_{\pi'}(e')$ we have that $\rho(f', y_{k+1})$ is a balanced reversal (type A.2s if $f' \neq y_{j-1}$, type C.3 otherwise). The required conditions are fulfilled since both types need $y_{k+1} > Pred_{\pi'}(f')$ and this is true by the definition of the cutpoint e', which is $Pred_{\pi'}(f')$; and in type C.3 we moreover need $f' > Next_{\pi'}(y_{k+1})$ and this is true too by the definition of the cutpoint e', since $f' > y_{k+1} > Next_{\pi'}(y_{k+1})$. The resulting permutation π'' has $v''_{k+1} = v'_{k+1} = v_j$ and the pinnacles are in increasing order.

Using the previous approach for each $k = 0, 1, \ldots, p$, we obtain a permutation still denoted π whose pinnacles are in increasing order and whose dells are identical to those of Id_S, and in increasing order. Each k requires 0 or 2 balanced reversals, depending whether v_{k+1} is already correct or not, so that at most $2p+2$ reversals are performed. $\quad\square$

Algorithm 4 Permutation sorting by balanced reversals: Step 2

Input: A permutation $\pi \in S_n$ with pinnacle set S of cardinality p. The pinnacles of π are increasingly ordered.

Output: The permutation π with pinnacles still in increasing order, and whose dells have become equal to the dells of Id_S, in increasing order (Proposition 17)

```
 1: for k = 0 to p do
 2:     w ← min({1, 2, . . . , n} − S − {v₁, . . . , vₖ})              // x is the lowest wished dell
 3:     if ∃vⱼ such that w = Predπ(vⱼ) then
 4:         π ← ApplyLemma15(π, vⱼ, vₖ₊₁)
 5:     else
 6:         if ∃vⱼ such that w = Nextπ(vⱼ) then
 7:             π ← ApplyLemma16(π, vⱼ, vₖ₊₁)
 8:         else
 9:             let vⱼ = w                                              //w is a dell
10:             if j ≠ k + 1 then
11:                 e ← cutAπ(vₖ₊₁, vⱼ, yⱼ); π′ ← π · ρ(vₖ₊₁, e); π″ ← π′
12:                 if k + 1 ≠ j − 1 then
13:                     e′ ← cutAπ′(yₖ₊₁, vⱼ, yⱼ₋₁); π″ ← π′ · ρ(Nextπ′(e′), yₖ₊₁);
14:                 end if
15:                 π ← π″
16:             end if
17:         end if
18:     end if
19: end for
20: Return π
```

Table 2 shows an example.

4.3   Step 3: Move the remaining elements towards the place they occupy in the canonical permutation

It remains to move in π the elements from each ascending and each descending set towards the end of the permutation.

Proposition 18. *Let $\pi \in S_n$ be a permutation with pinnacles $y_1 < y_2 \cdots < y_p$, and dells $v_1 < v_2 < \cdots < v_{p+1}$ which are the $p + 1$ lowest values in $\{1, 2, \ldots, n\} \setminus S$. There is a sequence R_3 of at most $2(n - 2p) - 1$ balanced reversals that transforms π into Id_S.*

Proof. This is done as follows. By hypothesis, v_{p+1} is smaller than all the elements from each ascending and each descending set, since the dells are the smallest elements that are not pinnacles.

   a) As long as $Next_\pi(v_{p+1}) < y_p$, we use the following trick to move $Next_\pi(v_{p+1})$ towards $D_\pi(y_p, v_{p+1})$ without changing the rest of π. Let π^{rev} be the permutation obtained from π by reversing the whole π. Lemma 15 i) may be applied to π^{rev} with $i =$

$q = p+1$ in order to move $Pred_{\pi^{rev}}(v_{p+1})$ towards $A_{\pi^{rev}}(v_{p+1}, y_p)$ using two balanced reversals (recall that $y_p > Next_\pi(v_{p+1}) = Pred_{\pi^{rev}}(v_{p+1})$ by the hypothesis above). Now, if we apply the balanced reversals with the same endpoints in π (without reversing the whole permutation), we obtain that $Next_\pi(v_{p+1})$ is moved towards $D_\pi(y_p, v_{p+1})$ without changing the rest of π. The resulting permutation is still called π and we continue. When the process is finished, each $t \in A_\pi(v_{p+1}, y_{p+1})$ satisfies $t > y_p$.

b) For each $i \leqslant p$, as long as $Next_\pi(v_i) \neq y_i$, use Lemma 16 to move $Next_\pi(v_i)$, which is smaller than y_i and thus smaller than y_p, towards $D_\pi(y_p, v_{p+1})$, without changing the rest of π. More precisely, item i) in the lemma is used, by the minimality of v_{p+1}. When this step is finished, we have $A_\pi(v_i, y_i) = \emptyset$ for all i with $1 \leqslant i \leqslant p$.

c) If $Next_\pi(y_p) \neq v_{p+1}$, the reversal $\rho(Next_\pi(y_p), v_{p+1})$ is balanced (type B.3s), since $Next_\pi(y_p) < y_p < Next_\pi(v_{p+1})$ by the constructions in the two previous items above, and $v_{p+1} < Pred_\pi(Next_\pi(y_p)) = y_p$. Then, in the new permutation still denoted π, $D_\pi(y_p, v_{p+1}) = \emptyset$.

d) For each $i \leqslant p$, as long as $Pred_\pi(v_i) \neq y_{i-1}$, use Lemma 15 to move $Pred_\pi(v_i)$ towards $A_\pi(v_{p+1}, y_{p+1})$ (item i) in the lemma) without changing the rest of π. When this step is finished, we have $D_\pi(v_i, y_i) = \emptyset$ for all i with $0 \leqslant i \leqslant p$.

It is easy to see that the result of these transformations is Id_S. Indeed, item b) ensures that v_i immediately precedes y_i, for each pinnacle y_i, $1 \leqslant i \leqslant p$. Once step b) is performed, the elements in $D_\pi(y_p, v_{p+1})$ are smaller than y_p whereas by item a) those in $A_\pi(v_{p+1}, y_{p+1})$ (if any) exceed y_p. The reversal in item c) thus only makes y_p adjacent to v_{p+1} by concatenating the elements in $D_\pi(y_p, v_{p+1})$ to those in $A_\pi(v_{p+1}, y_{p+1})$. Finally, item d) ensures that each y_i is adjacent to each v_{i+1} for $0 \leqslant i \leqslant p-1$, by successively inserting each element in $D_\pi(y_i, v_{i+1})$ into $\{v_{p+1}\} \cup A_\pi(v_{p+1}, y_{p+1})$.

As a consequence, all elements but the p pinnacles and the $p+1$ dells are possibly moved in items a), b) and d) using Lemma 15 or Lemma 16, thus performing two reversals per element. Since in item c) only one reversal is performed, the total number of reversals is at most $2(n - p - (p+1)) + 1 = 2(n - 2p) - 1$. $\qquad\square$

See the example in Table 2.

Proof of Theorem 7 The sequence R obtained by concatenating the sequences R_1, R_2, R_3 issued from Propositions 14, 17 and respectively Proposition 18 transforms π into Id_S as shown by these propositions. The number of balanced reversals in R needs to identify three cases:

- when $p = 1$, Steps 1, 2, 3 respectively take at most 0, $2p + 2$ and $2(n - 2p) - 1$ reversals, so the total number is $0 + (2p + 2) + (2n - 4p - 1) = 2n - 2p + 1$, so that with $p = 1$ we have $2n - 1$ reversals.

Algorithm 5 Permutation sorting by balanced reversals: Step 3
___
Input: A permutation $\pi \in S_n$ with pinnacle set S of cardinality p. The pinnacles of π are increasingly ordered. The dells of π are the same as those of Id_S.

Output: The permutation Id_S, obtained after placing into their correct places the elements of π not yet correctly placed (Proposition 18)
___
1: **while** $Next_\pi(v_{p+1}) < y_p$ **do**
2:     $u \leftarrow Next_\pi(v_{p+1}); e \leftarrow cutD_\pi(u, y_p, v_{p+1}); \pi' \leftarrow \pi \cdot \rho(e, u); \pi'' \leftarrow \pi' \cdot \rho(v_{p+1}, e); \pi \leftarrow \pi''$
    //item a)
3: **end while**
4: **for** $i = 1$ to p **do**
5:     **while** $Next_\pi(v_i) \neq y_i$ **do**
6:       $\pi \leftarrow ApplyLemma16(\pi, v_i, v_{p+1})$ //item b)
7:     **end while**
8: **end for**
9: **if** $Next_\pi(y_p) \neq v_{p+1}$ **then**
10:     $\pi \leftarrow \pi \cdot \rho(Next_\pi(y_p), v_{p+1})$ //item c)
11: **end if**
12: **for** $i = 1$ to p **do**
13:     **while** $Pred_\pi(v_i) \neq y_{i-1}$ **do**
14:       $\pi \leftarrow ApplyLemma15(\pi, v_i, v_{p+1})$ //item d)
15:     **end while**
16: **end for**
17: Return π
___

- when $p = 2$, Steps 1, 2, 3 respectively take at most 1, $2p + 2$ and $2(n - 2p) - 1$ reversals, so the total number is $1 + (2p + 2) + (2n - 4p - 1) = 2n - 2p + 2$, so that with $p = 2$ we have $2n - 2$ reversals.

- when $p \geqslant 3$, Steps 1, 2, 3 respectively take at most $2p - 4$, $2p + 2$ and $2(n - 2p) - 1$ reversals, so the total number is $(2p - 4) + (2p + 2) + (2n - 4p - 1) = 2n - 3$ reversals.

The number of reversals is therefore bounded by $2n - \min\{p, 3\}$ in all cases if $p \geqslant 1$.☐

Remark 19. In the case where $p = 0$, Steps 1 and 2 in the algorithm are not performed. Step 3 reduces to items c) and d) which perform 1 and respectively at most $2(n - 1)$ reversals. The total number of reversals is thus upper bounded by $2n - 1$ in this case.

5   Running time

The algorithm for transforming π in Id_S is the concatenation of Algorithms 1, 4 and 5 above, and is called Algorithm **BalancedSorting**. In this section we briefly show that **BalancedSorting** may be implemented in $O(n \log n)$.

The operations that need to be efficiently implemented, *i.e.* in $O(\log n)$ each, are easily identified.

- In Algorithm 1: successively computing the first, second etc. minimum of the initial set S of pinnacles (lines 1 and 19), computing $cutA_\pi$ for a given element and a given pinnacle (lines 15 and 21), finding the leftmost pinnacle larger than a given value and with index larger than a given value (lines 4 and 12), computing $Next_\pi(x)$ for a given element x (lines 15 and 21), performing the reversal between two given elements (lines 15 and 21).

- In addition, in Algorithm 4: computing the first, second etc. minimum of the set of wished dells (line 2), computing $cutD_\pi$ for a given element and a given pinnacle (line 7), computing $Pred_\pi(x)$ for a given element x (lines 3, 4, 7), deciding whether a given element is adjacent to a dell (lines 3 and 6).

- Algorithm 5 has no supplementary requirements.

These operations require to combine several efficient data structures that allow both a rapid access to the information and an efficient update. In particular, performing a reversal is a sensitive issue since it modifies the places of many elements, and - in the precise case we study here - swaps ascending and descending sets of the reversed block. In order to avoid recording all these changes one by one, the solutions proposed in literature in order to perform a (not necessarily balanced) reversal use three types of approaches. They are due to Kaplan and Verbin [8] (needs $O(\sqrt{n \log n})$ time to perform a reversal whose endpoints are known), Han [7] (needs $O(\sqrt{n})$ time for the same task) and Rusu [10] (needs $O(\log n)$ time for the same task). The latter one, that we choose for efficiency reasons, uses so-called *log-lists*. Log-lists may be assimilated, with a view to simplification, to double-linked lists in which a collection of operations may be performed in $O(\log n)$. The ones we are interested in here are: given (a pointer on) x, compute (the pointers on) $Pred_\pi(x)$ and $Next_\pi(x)$; change the sign (positive or negative) of all the elements of a sublist; perform a reversal (and update the structure).

The data structure we propose for the implementation of our algorithm combines log-lists, binary search trees (BSTs, for short), arrays and pointers. The shape of the permutation π is stored in a log-list L. For each pinnacle y_i in the shape, two pointers toA and toD go towards the roots of two BSTs. BST toA (respectively toD) contains the elements in $A_\pi(v_i, y_i)$ (respectively in $D_\pi(y_i, v_{i+1})$) represented as pairs (π_a, a). The order between pairs is defined as the standard (increasing) order between their left values (the elements). This implies that the pairs are also ordered according to the increasing order of their right values (the indices) in toA, respectively according to the decreasing order of their right values (the indices) in toD. Both of them, elements and indices, are used by the algorithm. An array P of pointers contains, in this order, the pointers to the pinnacles in L in increasing order of the pinnacles, followed by pointers to the wished dells (the dells of Id_S), in increasing order of the dells. The wished dells belong either to L or to one of the $2p$ BSTs pointed by the pointers toA, toD of each pinnacle.

With this data structure, we have:

- All the operations except finding the leftmost pinnacle larger than a given value and with index larger than a given value (lines 4 and 12 of Algorithm 1) take $O(\log n)$

time each. This is quite easy, given the abovementioned properties of a log-list and those of a BST (searching a value in a BST, cutting a BST, merging two BSTs take $O(\log n)$ time). It is important to notice that a reversal is performed on the shape of the permutation (thus in the log-list L), but: 1) it cuts either toA or toD at each endpoint of the reversed block (except when the endpoint is a pinnacle or a dell) and recombines the resulting BSTs (also modifying the dells at the endpoints if needed) once the reversal is performed, both in $O(\log n)$ time; 2) it swaps the roles of toA and toD for each pinnacle situated strictly inside the reversed block, and thus each reversal must be followed by a sign change (in $O(\log n)$ time) in the reversed block of L (whose meaning is that when the value of the pinnacle is positive, toA and toD correspond respectively to the ascending and descending sets neighboring the pinnacle, whereas when the value is negative their roles are swapped).

- The operation of finding the leftmost pinnacle larger than a given value and with index larger than a given value (lines 4 and 12 of Algorithm 1) cannot be implemented in $O(\log n)$ time per operation. Instead, it may be shown that all these operations (in lines 4 and 12 of Algorithm 1) may be implemented altogether in $O(n \log n)$ time.

  To this end, one has to remember that in Lemma 11 when Case 2 (tested in lines 4 and 12) is applied to y_i, one finds a first index $h > i$ such that $v_1 < y_h$. Then we reverse the block with endpoints v_1 and v_h. Again as proved in Lemma 11, the values h^q computed similarly with h in the next executions of the *while* loop (lines 9-17 in Algorithm 1) belong to the block of π situated between v_1 and y_h, and become smaller at each execution. That means each reversal due to Case 2 reverses a proper prefix of the block resulting after the latest reversal. Then it is sufficient to perform in $O(n \log n)$ time a pre-treatment of the block of the (initial) permutation π with endpoints v_1 and y_h (once h is known). This pre-treatment affects to each dell v_j with $j < i$ a pointer to the leftmost pinnacle $y_{h'}$ such that $y_{h'} > v_j$ and $i < h' < h$, and symmetrically affects to each dell v_j with $j > i$ a pointer to the rightmost pinnacle $y_{h'}$ such that $y_{h'} > v_j$ and $h' < i$. By the previous considerations, the reversals performed by the algorithm do not affect the meaning of the pointers affected to the dells v_1^q obtained during the execution of Case 2. These pointers, computed independently from the data structure presented above, may be stored as additional information in the cells representing the dells in the log-list. They allow to find the pinnacles $y_{h^q}^q$ in $O(1)$ time each, and thus the dells v_h needed by Algorithm 1 (line 13) in $O(\log n)$ time each.

6 Conclusion

We have shown in this paper that the BALANCED SORTING problem has a solution using at most $2n - \min\{p, 3\}$ balanced reversals (when $p \geqslant 1$), for each permutation of n elements. This is an upper bound, but many permutations may be sorted using less balanced reversals. Then we can ask:

Question 1. (MINIMUM BALANCED SORTING) Given $\pi \in S_n$ with pinnacle set S, find the minimum number of balanced reversals needed to transform π into Id_S.

A similar question may be asked when transforming a permutation π into a permutation π' with the same pinnacle set:

Question 2. (MINIMUM BALANCED TRANSFORMATION) Given $\pi, \pi' \in S_n$ with the same pinnacle set S, find the minimum number of balanced reversals needed to transform π into π'.

This question is probably different from the previous one, since the left-invariance is not ensured with respect to the canonical permutation. Indeed, examples exist where using the canonical permutation as intermediate permutation in the transformation of π into π' (therefore solving MINIMUM BALANCED SORTING twice) does not allow to reach the minimum number of balanced reversals obtained when solving MINIMUM BALANCED TRANSFORMATION. An example is given by $\pi = (8\,5\,\underline{2}\,\overline{6}\,\underline{4}\,\overline{7}\,\underline{1}\,3\,9)$ and $\pi' = (8\,5\,\underline{1}\,\overline{7}\,\underline{4}\,\overline{6}\,\underline{2}\,3\,9)$, where one balanced reversal of type (C2), namely $\rho(2,1)$, allows us to transform π into π' without using the canonical permutation as an intermediate permutation.

A related problem is raised by Example 6. Even when a set is admissible as the pinnacle set of a permutation, the order of the pinnacles in the permutation is important, since some orders may be impossible to respect. Identifying these orders could be a way to better target the balanced reversals yielding a minimum sorting.

Question 3. Let S be an admissible pinnacle set. Give a characterization of the total orders σ on S such that a permutation $\pi \in S_n$ exists whose sequence of pinnacles read from left to right is exactly σ.

Note that Davis et al. [5] study the number of permutations on n elements with a given admissible set S, and give recursive formulas for it. This is a related question, which involves however supplementary combinatorial aspects related to the elements not belonging to S.

References

[1] Marcelo Aguiar, Kathryn Nyman, and Rosa Orellana. New results on the peak algebra. *Journal of Algebraic Combinatorics*, 23(2):149–188, 2006.

[2] Louis J Billera, Samuel K Hsiao, and Stephanie van Willigenburg. Peak quasisymmetric functions and eulerian enumeration. *Advances in Mathematics*, 176(2):248–276, 2003.

[3] Sara Billey, Krzysztof Burdzy, and Bruce E Sagan. Permutations with given peak set. *Journal of Integer Sequences*, 16(6), 2013.

[4] Pierre Bouchard, Hungyung Chang, Jun Ma, Jean Yeh, and Yeong-Nan Yeh. Value-peaks of permutations. *The Electronic Journal of Combinatorics*, 17(1):#R46, 2010.

[5] Robert Davis, Sarah A Nelson, T Kyle Petersen, and Bridget E Tenner. The pinnacle set of a permutation. *Discrete Mathematics*, 341(11):3249–3270, 2018.

[6] Alexander Diaz-Lopez, Pamela E Harris, Erik Insko, and Mohamed Omar. A proof of the peak polynomial positivity conjecture. *Journal of Combinatorial Theory, Series A*, 149:21–29, 2017.

[7] Yijie Han. Improving the efficiency of sorting by reversals. *BIOCOMP*, 6:406–409, 2006.

[8] Haim Kaplan and Elad Verbin. Efficient data structures and a new randomized approach for sorting signed permutations by reversals. In *CPM 2003*, pages 170–185, 2003.

[9] Sergey Kitaev, Toufik Mansour, and Jeffrey B Remmel. Counting descents, rises, and levels, with prescribed first element, in words. *Discrete Mathematics and Theoretical Computer Science*, 10(3), 2008.

[10] Irena Rusu. Log-lists and their applications to sorting by transpositions, reversals and block-interchanges. *Theoretical Computer Science*, 660:1–15, 2017.

[11] Manfred Schocker. The peak algebra of the symmetric group revisited. *Advances in Mathematics*, 192(2):259–309, 2005.