Pattern matching for $(213, 231)$-avoiding permutations

Both Emerite Neou⋆[1] Romeo Rizzi[2] and Stéphane Vialette[1]

[1] Université Paris-Est, LIGM (UMR 8049), CNRS, UPEM, ESIEE Paris, ENPC, F-77454, Marne-la-Valle, France
`{neou,vialette}@univ-mY.fr`
[2] Department of Computer Science, Universit degli Studi di Verona, Italy
`romeo.rizzi@univr.it`

Abstract. Given permutations $\sigma \in S_k$ and $\pi \in S_n$, the Permutation Pattern (PP) problem is to decide whether π contains σ as an order-isomorphic pattern. We give a linear-time algorithm in case both π and σ avoid 213 and 231. For the special where only σ avoids 213 and 231, we present a $O(kn^2)$ time algorithm. We extend our research to $(213, 231)$-avoiding bivincular pattern and present a $O(kn^3)$ time algorithm.

1 Introduction

A permutation π is said to contain another permutation σ, in symbols $\sigma \preceq \pi$, if there exists a subsequence of entries of π that has the same relative order as σ, and in this case σ is said to be a *pattern* of π. Otherwise, π is said to avoid the permutation σ. For example a permutation contains the pattern 123 (resp. 321) if it has an increasing (resp. decreasing) subsequence of length 3. Here, note that members need not actually to be consecutive, merely increasing (resp. descreasing). As another example, 6152347 contains 213 but not 231. During the last decade, the study of the pattern containment on permutations has become a very active area of research and a whole annual conference (Permutation Patterns) is now devoted to this topic.

We consider here the so-called *pattern containment* problem (also sometimes referred to as the *pattern involvement problem*): Given two permutations σ and π, this problem is to decide whether $\sigma \preceq \pi$ (the problem is ascribed to Wilf in [**?**]). The permutation containment problem is **NP**-hard [**?**]. It is, however, polynomial-time solvable by brute-force enumeration if σ has bounded size. Improvements to this algorithm were presented in [**?**] and [**?**], the latter describing a $O(|\pi|^{0.47|\sigma|+o(|\sigma|)})$ time algorithm. Bruner and Lackner [**?**] gave a fixed-parameter algorithm solving the pattern containment problem with an exponential worst-case runtime of $O(1.79^{\mathrm{run}(\pi)})$, where $\mathrm{run}(\pi)$ denotes the number of alternating runs of π. (This is an improvement upon the $O(2^{|\pi|})$ runtime required by brute-force search without imposing restrictions on σ and π.) A recent major

⋆ On a Co-tutelle Agreement with the Department of Mathematics of the University of Trento step was taken by Marx and Guillemot [?]. They showed that the permutation containment problem is fixed-parameter tractable for parameter $|\sigma|$.

A few particular cases of the pattern containment problem have been attacked successfully. The case of increasing patterns is solvable in $O(|\pi| \log \log |\sigma|)$ time in the RAM model [?], improving the previous 30-year bound of $O(|\pi| \log |\sigma|)$. (The algorithm also improves on the previous $O(|\pi| \log \log |\pi|)$ bound.) Furthermore, the patterns 132, 213, 231, 312 can all be handled in linear-time by stack sorting algorithms. Any pattern of length 4 can be detected in $O(|\pi| \log |\pi|)$ time [?]. Algorithmic issues for 321-avoiding patterns containment has been investigated in [?]. The pattern containment problem is also solvable in polynomial-time for separable patterns [?,?] (see also [?] for LCS-like issues of separable permutations). Separable permutations are those permutations that contain neither 2413 nor 3142, and they are enumerated by the Schrder numbers (sequence A006318 in OEIS). To the best of our knowledge, separable permutations first arose in the work of Avis and Newborn [?], who showed that they are precisely the permutations which can be sorted by an arbitrary number of pop-stacks in series, where a pop-stack is a restricted form of stack in which any pop operation pops all items at once. (Notice that the separable permutations include as a special case the stack-sortable permutations, which avoid the pattern 231.)

There exist many generalisation of patterns that are worth considering in the context of algorithmic issues in pattern involvement (see [?] for an up-to-date survey). *Vincular patterns*, also called *generalized patterns*, resemble (classical) patterns, with the constraint that some of the letters in an occurrence must be consecutive. Of particular importance in our context, Bruner and Lackner [?] proved that deciding whether a vincular pattern σ of length k occurs in a longer permutation π is $W[1]$-complete for parameter k. *Bivincular patterns* generalize classical patterns even further than vincular patterns. Indeed, in bivincular patterns, not only positions but also values of elements involved in a matching may be forced to be adjacent

We focus in this paper on pattern matching issues for $(213, 231)$-avoiding permutations. The number of n-permutations that avoid both 213 and 231 is $t_0 = 1$ for $n = 0$ and $t_n = 2^{n-1}$ for $n \geq 1$ [?]. On an individual basis, the permutations that do not contain the permutation pattern 231 are exactly the *stack-sortable permutations* and they are counted by the Catalan numbers [?]. A stack-sortable permutation is a permutation whose elements may be sorted by an algorithm whose internal storage is limited to a single stack data structure. As for 213, it is well-known that if $\pi = \pi_1\pi \ldots \pi_n$ avoids 132, then its complement $\pi' = (n+1-\pi_1)(n+1-\pi_2) \ldots (n+1-\pi_n)$ avoids 312, and the reverse of π' avoids 213. From a combinatorial point of view, Bóna [?] showed the rather surprising fact that the cumulative number of occurrences of the classical patterns 231 and 213 are the same on the set of permutations avoiding 132, beside the pattern based statistics 231 and 213 do not have the same distribution on this set. *Almost avoidance* has also been considered; a permutation π almost avoids a set X of permutations if there is a way to remove a single element of π to get a permutation that avoids all elements in X and $L_n(X)$ denotes the set of permutations of length

n that almost avoid X. It is shown in [**?**] that $L_n(213, 231) = L_n(132, 231) = L_n(132, 312) = L_n(213, 312)$.

This paper is organized as follows.

2 Definition

A *permutation* of length n is a one-to-one function from an n-element set to itself. We write permutations as words $\sigma = \sigma_1\sigma_2 \ldots \sigma_n$, whose letters are distinct and uasually consist of the integers $12 \ldots n$. We let $\pi[i]$ stands for π_i. For the sake of convenience, we let $\pi[i : j]$ stand for $\pi_i\pi_{i+1} \ldots \pi_j$, $\pi[: j]$ stand for $\pi[1 : j]$ and $\pi[i :]$ stand for $\pi[i : n]$. As usual, we let S_n denote the set of all permutations of length n.

A permutation π is said to *contain* the permutation σ if there exists a subsequence of (not necessarily consecutive) entries of π that has the same relative order as σ, and in this case σ is said to be a *pattern* of π, written $\sigma \preceq \pi$. Otherwise, π is said to *avoid* the permutation σ. For example, the permutation $\pi = 391867452$ contains the pattern $\sigma = 51342$, as can be seen in the highlighted subsequence of $\pi = \mathbf{391}8\mathbf{674}52$ (or $\pi = \mathbf{391}8\mathbf{67}4\mathbf{5}2$ or $\pi = \mathbf{391}8\mathbf{67}45\mathbf{2}$). Each subsequence (91674, 91675, 91672) is called a *copy*, *instance*, or *occurrence* of σ. Since the permutation $\pi = 391867452$ contains no increasing subsequence of length four, π avoids 1234.

Suppose P is a set of permutations. We let $\mathrm{Av}_n(p)$ denote the set of all n-permutations avoiding each permutation in P. For the sake of convenience, we omit P's braces thus having e.g. $\mathrm{Av}_3(132, 4312)$ instead of $\mathrm{Av}_3(\{132, 4312\})$. A basic example is if $\pi = \pi_1\pi_2 \ldots \pi_n$ avoids 321, i.e., has no decreasing subsequence of length 3, then its reverse, $\pi' = \pi_n\pi_{n-1} \ldots \pi_1$ avoids 123, i.e., has no increasing subsequence of length 3.

An *ascent* of a permutation $\pi \in S_n$ is any position $1 \leq i < n$ where the following value is bigger than the current one. That is, if $\pi = \pi_1\pi_2 \ldots \pi_n$, then i is an ascent if $\pi_i < \pi_{i+1}$. For example, the permutation 3452167 has ascents (at positions) $1, 2, 5, 6$. Similarly, a *descent* is a position $1 \leq i < n$ with $\pi_i > \pi_{i+1}$, so every i with $1 \leq i < n$ either is an ascent or is a descent of π. Let a and d denote an ascend and a descend, respectively. The *stripe* s_π of a permutation $\pi \in S_n$ is the word $\mathrm{s}_\pi[1]\,\mathrm{s}_\pi[2] \ldots \mathrm{s}_\pi[n-1] \in \{a, d\}^{n-1}$ defined by $\mathrm{s}_\pi[i] = a$ if i an ascent in π and $\mathrm{s}_\pi[i] = d$ if i a descent in π. For example the stripe of the permutation $\pi = 981234765$ is $\mathrm{s}_\pi = ddaaaadd$. The stripe s_σ is a *pattern* of the strip s_π (or s_σ *occurs* of the strip s_π) if s_σ occurs in s_π as a subsequence.

A *bivincular pattern* (abbreviated BVP) σ of length k is a permutation in S_k written in two-line notation (that is the top row is $12 \ldots k$ and the bottom row is a permutation $\sigma_1\sigma_2 \ldots \sigma_k$). We have the following conditions on the top and bottom rows of σ:

– If the bottom line of σ contains $\underline{\sigma_i\sigma_{i+1} \ldots \sigma_j}$ then the letters corresponding to $\sigma_i\sigma_{i+1} \ldots \sigma_j$ in an occurrence of σ in a permutaton must be adjacent, whereas there is no adjacency condition for non-underlined consecutive letters. Moreover if the bottom row of σ begins with $\llcorner\sigma_1$ then any occurrence of σ in a permutation π must begin with the leftmost letter of π, and if the bottom row of σ begins with $\sigma_{k \lrcorner}$ then any occurrence of σ in a permutation π must end with the rightmost letter of π.

- If the top line of σ contains $\overline{i\, i+1 \ldots j}$ then the letters corresponding to $\sigma_i, \sigma_{i+1}, \ldots, \sigma_j$ in an occurrence of σ in a permutation must be adjacent in values, whereas there is no value adjacency restriction for non-overlined letters. Moreover, if the top row of σ begins with $\ulcorner 1$ then any occurrence of σ is a permutation π must begin with the smallest letter of π, and if top row of σ ends with $k^{\urcorner}$ then any occurrence of σ is a permutation π must end with the largest letter of π.

Given a motif σ we represent the elements underlined by the set X. An index i of an element is in X if and only if i and $i{+}1$ need to be adjacent ($\sigma[i]$ and $\sigma[i{+}1]$ are underlined). For example if we have $\underline{\sigma_i \sigma_{i+1} \ldots \sigma_j}$, then $i, i+1, \ldots, j-1 \in Y$. We represent the index that need to be overlined by the set Y. An index i is in Y if and only if i and $i{+}1$ need to be adjacent (i and $i{+}1$ are overlined). For example if we have For example if we have $\overline{i\, i+1 \ldots j}$, then $i, i+1, \ldots, j-1 \in Y$.

3 Both π and σ are $(213, 231)$-avoiding

This section is devoted to presenting a fast algorithm for deciding $\sigma \leq \pi$ in case both π and σ are $(213, 231)$-avoiding. We begin with an easy structure lemma.

Lemma 1 (Folklore). *The first element of any* $(213, 231)$*-avoiding permutation must be either* 1 *or* n*.*

Proof. Any other initial element would serve as a '2' in either a 231 or 213 with 1 and n as the '1' and '3' respectively.

The above lemma gains in interest in the form of the following corollaries.

Corollary 1. $\pi \in \mathrm{Av}_n(213, 231)$ *if and only if for* $1 \leq i \leq n$, $\pi[i:]$ *start either with the maximal element or the minimal element of* $\pi[i:]$*.*

The following lemma is central to all our algorithm.

Lemma 2. *Let* π *and* σ *be two* $(213, 231)$*-avoiding permutations, given a subsequence* t *of* π*,* t *is an occurrence of* σ *in* π *if and only if* s_t *is an occurrence of* s_σ *in* s_π*.*

Proof. The forward direction is obvious. We prove the backward direction by induction on the size of the motif σ. The base case is a motif of size 2. Suppose that we have the motif (12), thus the stripe of the motif is a. Let's t be a subsequence of π such that s_t is a occurrence of s_σ in s_π. Because we have $\mathrm{s}_t[1] = a$, the next element of $t[1]$ is π is superior to $t[1]$, so we have a match. Same idea goes of motif (21). Now, suppose that the lemma is true for every motif in $\mathrm{Av}_n((231, 213))$. Suppose that we have a subsequence t of π such that s_t is an occurrence of s_σ in s_π, then $\mathrm{s}_{t[2:]}$ is an occurrence of $\sigma[2:]$ in π. Moreover

$\sigma[2:] \in \mathrm{Av}_n((231, 213))$ so by hypothesis of induction $t[2:]$ is an occurrence of $\sigma[2:]$ in π. Moreover if $\mathrm{s}_\sigma[1]$ is ascend (descent), then by lemma 1 $\sigma[1]$ is the minimal (maximal) element of the motif. Furthermore s_t is an occurrence of s_σ in s_π , so $\mathrm{s}_t[1]$ is also ascend (descent) thus $t[1]$ is also the minimal (maximal) element of the subsequence t which complete our proof.

□

We are now ready to solve the pattern containment problem in case both π and σ are $(213, 231)$-avoiding.

Proposition 1. *Let π and σ be two $(213, 231)$-avoiding permutations. One can decide whether π contains σ in linear-time.*

Proof. According to Lemma 2 the problem reduces to deciding whether s_σ occurs as a subsequence in s_π. A straightforward greedy approach solves this issue in linear-time. Furthermore, in the light of Corollary 1, the effective construction of the two stripes is not required, so it is an online algorithm. □

4 σ only is $(231, 213)$-avoiding

This section focus on the pattern containment problem in case only the pattern σ avoids 231 and 213.

Proposition 2. *Let $\sigma \in \mathrm{Av}_k(213, 231)$ and $\pi \in S_n$. One can decide in $O(kn^2)$ time and $O(kn^2)$ space if π contains σ.*

For the proof and the algorithm we introduce a decomposition : we decompose the stripe of the motif into factor of same letter, and label those factor from right to left. We also need to add the last index into the first factor. For each factor, we denote the left most element by L. For example, given the motif 981237654, $\mathrm{F}(1) = 7654$, $\mathrm{L}(1) = 7$, $\mathrm{F}(2) = 123$, $\mathrm{L}(2) = 1$, $\mathrm{F}(3) = 98$ and $\mathrm{L}(9)$.

Proof (of Proposition 2). Given an ascent (descent) factor $\mathrm{F}(i)$ of a motif σ, if we want to match the suffix of σ starting by $\mathrm{F}(i)$, $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$ and given that we know every match of $\mathrm{F}(i-1)\cdots\mathrm{F}(1)$, the optimal match is to choose the match of $\mathrm{F}(i-1)\cdots\mathrm{F}(1)$ that minimize (maximize) the maximal (minimal) element of the match of $\mathrm{F}(i-1)\cdots\mathrm{F}(1)$ and that allow a match of factor $\mathrm{F}(i)$. In other word if we want to match $\sigma[\mathrm{L}(i):]$ with $\pi[j:]$ such that $\mathrm{L}(i)$ is matched with element at index j in π, the optimal way of matching $\sigma[\mathrm{L}(i-1):]$ is to choose the minimal (maximal) element at index j' of $\pi[j:]$ such that : (1) $\sigma[\mathrm{L}(i-1):]$ is matched with $\pi[j':]$ and element $\mathrm{L}(i-1)$ is matched with element at index j', (2) the factor $\mathrm{F}(i)$ is matched with $\pi[j, j'-1]$ and element $\mathrm{L}(i)$ is matched with element at index j, (3) Every elements of the match of (2) if inferior (superior) to the minimal (maximal) element of the match of (1). Indeed when we will match $\mathrm{F}(i+1)\,\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$, every element of $\mathrm{F}(i+1)$ must be matched to element superior (inferior) to every element matched element of $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$, particularly its must be superior (inferior) to the maximal (minimal) element of the match of $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$.

Remark 1. Given an ascent (descent) factor $\mathrm{F}(i)$, the maximal (minimal) element of $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$ is $\mathrm{L}(i-1)$.

Remark 2. An ascent (descent) factor $\mathrm{F}(i)$ is a match in $\pi[j:j']$ if and only if there exists an increasing (decreasing) subsequence in $\pi[j:j']$ of size superior or equal to the size of $\mathrm{F}(i)$.

Consider the following problem :
$LM(i,j)$ = the optimal match of $\mathrm{L}(i-1)$ if there exists a match of $\sigma[\mathrm{L}(i):]$ with $\pi[j:]$ and element $\mathrm{L}(i)$ is matched with element at index j.

This problem can be solve by induction.

BASE :

$$LM(1,j) = \begin{cases} MIN_{j<j'}\{0\} \cup \{\pi[j']|\ j' \text{ such that} & \text{if } \mathrm{s}_\sigma[\mathrm{L}(1)] = a \\ \ |\,\mathrm{F}(1)| \leq LIS(j,j',\pi[j'])\} & \\ & \\ MAX_{j<j'}\{0\} \cup \{\pi[j']|\ j' \text{ such that} & \text{if } \mathrm{s}_\sigma[\mathrm{L}(1)] = d \\ \ |\,\mathrm{F}(1)| \leq LDS(j,j',\pi[j'])\} & \end{cases}$$

STEP :

$$LM(i,j) = \begin{cases} MIN\{0\} \cup I(i,j) & \text{if } \mathrm{s}_\sigma[\mathrm{L}(i)] = a \\ MAX\{0\} \cup D(i,j) & \text{if } \mathrm{s}_\sigma[\mathrm{L}(i)] = d \end{cases}$$

With $I(i,j)$ $(D(i,j))$ the set of element such that if $j' \in I(i,j)$ then there exists a match of $\sigma[\mathrm{L}(i-1):]$ in $\pi[j':]$ and the element $\mathrm{L}(i-1)$ is matched with element at index j' and there is a match of $\mathrm{F}(i)$ in $\pi[j:j'-1]$ and the element $\mathrm{L}(i)$ is matched with element at index j and every element is inferior (superior) to the match of $\sigma[\mathrm{L}(i-1):]$ in $\pi[j':]$. Formally we define $I(s,i)$ and $D(s,i)$ such as:

$$I(i,j) = \{j'|j<j' \text{ and } LM(i-1,j') \neq 0 \text{ and } |\,\mathrm{F}(i)| \leq LIS(j,j'-1,\mathrm{L}(i-1))\}$$

$$D(i,j) = \{j'|j<j' \text{ and } LM(i-1,j') \neq 0 \text{ and } |\,\mathrm{F}(i)| \leq LDS(j,j'-1,\mathrm{L}(i-1))\}$$

With LIS(i,j,k) (LDS(i,j,k)) is the longest increasing (decreasing) sequence in π starting at i and ending at j, with every element of this sequence inferior (superior) to k. LIS and LDS can be computed in $O(|\pi|^2 * log(log(|\pi|)))$ (see [**?**]).

In the base case, we are looking for a match for the first factor. Each factor is either ascent or descent. If the factor is an ascent (descent) we have to find an increasing (decreasing) subsequence in the text of same size or longest that the size of the factor, and to find the optimal solution we must assure that the last element of the sequence is minimal (maximal).
For the induction, it is the same idea except that we must assure that every element of the subsequence is superior (inferior) to the maximum (minimal) element of the rest of the match ie inferior (superior) to the minimal (maximal)

element of the rest of the match which is given by LM of the previous factor.

There exists a match of σ in π if and only if there exists a $LM(n,i) \neq 0$ for $1 \leq i \leq |\pi|$, with n the number of factor in σ. Moreover the basic case can be computed in $O(|\pi|^2)$ and the induction on $O(|\sigma||\pi|^2)$. □

5 (213, 231)-avoiding bivincular patterns

This section is devoted to presenting a polynomial-time algorithm for deciding whether a $(213, 231)$-avoiding bivincular pattern occurs in a permutation.

Proposition 3. *Let σ be a $(213, 231)$-avoiding bivincular pattern of length k and $\pi \in S_n$. One can decide in $O(kn^3)$ time and $O(kn^3)$ space if π contains σ.*

Proof. We consider the following problem :
Given a bivincular motif σ^+ with $\sigma \in \mathrm{Av}_{n_\sigma}(231, 213)$, and a text $\pi \in \mathrm{Av}_{n_\pi}(231, 213)$, i,j , $i < j$.

$$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(i,j) = \begin{cases} true & \text{If } \sigma^+[i:] \text{ is a motif} \\ & \text{with every element} \\ & [\min,\max] \\ >>>>>>>\ f9bc0a20a19c155782e9e48e9d01a9fb6f3718ac & \\ false & otherwise \end{cases}$$

$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(i,j)$ is closed under induction. it can be solved by means of the following relations:

BASE :
if $n_\sigma \notin X$ and $n_\sigma \notin Y$:

$$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(n_\sigma,j) = \begin{cases} true & \text{if } \min < \pi[j] < \max \\ false & otherwise \end{cases}$$

if $n_\sigma \in X$ and $n_\sigma \notin Y$:

$$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(n_\sigma,n_\pi) = \begin{cases} true & \text{if } \min < \pi[n_\pi] < \max \\ false & otherwise \end{cases}$$

if $n_\sigma \notin X$ and $n_\sigma \in Y$ and $\sigma[n_\sigma]$ is the maximal element:

$$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(n_\sigma,j) = \begin{cases} true & \text{if } \min < \pi[j] < \max \text{ and} \\ & \pi[j] \text{ is the maximal element} \\ false & otherwise \end{cases}$$

if $n_\sigma \in X$ and $n_\sigma \in Y$ and $\sigma[n_\sigma]$ is the maximal element:

$$\mathrm{PM}^{\pi,\min,\max}_{\sigma^+}(n_\sigma,n_\pi) = \begin{cases} true & \text{if } \min < \pi[n_\pi] < \text{maxand} \\ & \pi[n_\pi] \text{ is the maximal element} \\ false & otherwise \end{cases}$$

STEP :

$$\mathrm{PM}_{\sigma^+}^{\pi,\min,\max}(i,j) = \begin{cases} \bigcup_{j<k} \mathrm{PM}_{\sigma^+}^{\pi,\pi[j],\max}(i+1,k) & \text{if } \mathrm{s}_\sigma[i] = a \text{ and } i \notin X \\ & \text{and } i \notin Y \\ & \text{and } \min < \pi[j] < \max \\ \bigcup_{j<k} \mathrm{PM}_{\sigma^+}^{\pi,\min,\pi[j]}(i+1,k) & \text{if } \mathrm{s}_\sigma[i] = d \\ & \text{and } i \notin X \text{ and } i \notin Y \\ & \text{and } \min < \pi[j] < \max \\ \mathrm{PM}_{\sigma^+}^{\pi,\pi[j],\max}(i+1,j+1) & \text{if } \mathrm{s}_\sigma[i] = a \\ & \text{and } i \in X \text{ and } i \notin Y \\ & \text{and } \min < \pi[j] < \max \\ \mathrm{PM}_{\sigma^+}^{\pi,\min,\pi[j]}(i+1,j+1) & \text{if } \mathrm{s}_\sigma[i] = d \\ & \text{and } i \in X \text{ and } i \notin Y \\ & \text{and } \min < \pi[j] < \max \\ \mathrm{PM}_{\sigma^+}^{\pi,\pi[j],\max}(i+1,\pi^{-1}[\pi[j]+1]) & \text{if } \mathrm{s}_\sigma[i] = a \\ & \text{and } i \notin X \text{ and } i \in Y \\ & \text{and } \min < \pi[j] < \max \\ & \text{and } \pi^{-1}[\pi[j]+1] > j \\ \mathrm{PM}_{\sigma^+}^{\pi,\min,\pi[j]}(i+1,\pi^{-1}[\pi[j]-1]) & \text{if } \mathrm{s}_\sigma[i] = d \\ & \text{and } i \notin X \text{ and } i \in Y \\ & \text{and } \min < \pi[j] < \max \\ & \text{and } \pi^{-1}[\pi[j]-1] > j \\ \mathrm{PM}_{\sigma^+}^{\pi,\pi[j],\max}(i+1,j+1) & \text{if } \mathrm{s}_\sigma[i] = a \\ & \text{and } i \in X \text{ and } i \in Y \\ & \text{and } \pi[j]+1 = \pi[j+1] \\ & \text{and } \min < \pi[j] < \max \\ \mathrm{PM}_{\sigma^+}^{\pi,\min,\pi[j]}(i+1,j+1) & \text{if } \mathrm{s}_\sigma[i] = d \\ & \text{and } i \in X \text{ and } i \in Y \\ & \text{and } \pi[j]-1 = \pi[j+1] \\ & \text{and } \min < \pi[j] < \max \\ \mathit{False} & \mathit{Otherwise} \end{cases}$$

At each step (i,j), If $i \notin X$ and $i \notin Y$, we match the current element of σ ($\sigma[i]$) with the current element of π ($\pi[j]$), if possible. Then we match $\sigma[i+1:]$ with every suffixes π starting after j.

If $i \in X$ and $i \notin Y$, we match $\sigma[i]$ to $\pi[j]$, then we match $\sigma[i+1:]$ with $\pi[j+1:]$ with the condition that $\sigma[i+1]$ is matched to $\pi[j+1]$.

If $i \notin X$, $i \in Y$ and i is a descent, we match $\sigma[i]$ to $\pi[j]$, then we match $\sigma[i+1:]$ with $\pi[\pi^{-1}[\pi[j]-1]:]$ with the condition that $\sigma[i+1]$ is matched to the element $\pi[j]-1$.

If $i \notin X$, $i \in Y$ and i is an ascent, we match $\sigma[i]$ to $\pi[j]$, then we match $\sigma[i+1:]$ with $\pi[\pi^{-1}[\pi[j]+1]:]$ with the condition that $\sigma[i+1]$ is matched to the element $\pi[j]+1$.

If $i \in X$, $i \in Y$ and i is a descent, we match $\sigma[i]$ to $\pi[j]$, then we match $\sigma[i+1:]$ with $\pi[j+1:]$ with the condition that $\sigma[i+1]$ is matched to $\pi[j+1]$, and that $\pi[j+1] = \pi[j]-1$.

If $i \in X$, $i \in Y$ and i is an ascent, we match $\sigma[i]$ to $\pi[j]$, then we match $\sigma[i+1:]$ with $\pi[j+1:]$ with the condition that $\sigma[i+1]$ is matched to $\pi[j+1]$, and that $\pi[j+1] = \pi[j]+1$.

There are two cases to consider searching for a match :

- If $n_\sigma \in X$ there exists a match if and only if there is a match starting with the first element of the text:
  $\bigcup_{0<\min<\max<n_\pi} \mathbb{P}_{\sigma,\pi,\min,\max}(0,0)$ is true.
- If $n_\sigma \notin X$ there exists a match if and only if there exists a match stating at any index:
  $\bigcup_{0<k<n_\pi, 0<\min<\max<n_\pi} \mathbb{P}_{\sigma,\pi,\min,\max}(0,k)$ is true.

¿¿¿¿¿¿¿ f9bc0a20a19c155782e9e48e9d01a9fb6f3718ac

6 Longest Subsequence Avoiding (231,213) for a Permutation

In this section we present an algorithm to solve the problem of the longest subsequence avoiding (231,213). To do so, we need the set of element that are matched to an ascent and the set of element that are matched to a descent. For a permutation π of size n, we define $P(\pi) = \{i|\, \mathrm{s}_\pi[i] = a\} \cup \{n\}$ and $M(\pi) = \{i|\, \mathrm{s}_\pi[i] = d\} \cup \{n\}$.

Proposition 4. *If s is a longest subsequence avoiding (213,231) with last element at index f in π then $P(\pi)$ is a longest increasing subsequence with last* *element at index f and $M(\pi)$ is a longest decreasing subsequence with last element at index f.*

Proof. Let's s be a longest subsequence avoiding (213,231) with last element at index f in π, suppose that $P(\pi)$ is not a longest increasing subsequence with last element at index f. Let's s_m be a longest increasing subsequence with last element f. Thus $|s_m| > |P(\pi)|$, clearly the sequence $s_m \cup M(\pi)$ is avoiding (213,231) and is longer than s witch is not possible. The same idea goes for the demonstration that $M(\pi)$ is the longest decreasing subsequence.

Proposition 5. *Given a permutation π, finding the longest subsequence avoiding (231,213) can be done in $O(|\pi| log(log(|\pi|)))$ time and in $O(n)$ space.*

Proof. The proposition **??** lead to algorithm where we have to compute longest increasing and decreasing subsequence ending at every index. Then finding the maximum sum of longest increasing and decreasing subsequence ending at the same index. Computing the longest increasing and decreasing can be done in $O(|\pi| * log(log(|\pi|)))$ time and $O(n)$ space (see [?]), then finding the maximum can be done in linear time.

7 Longest Subsequence Avoiding (231,213) Common for Two Permutation

In this section we present an algorithm to find the longest common subsequence avoiding (231,213) between two permutations.

Proposition 6. *Given two permutation π_1 and π_2, the longest common subsequence avoiding (231,213) can be compute in $O(|\pi_1|^3|\pi_2|^3)$ time and space.*

Proof. Consider the following problem, that compute the longest stripe common to π_1 and π_2. Given two permutation π_1 and π_2.

$\mathrm{LCS}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1, i_2) = \max\{\ |s|\ |$ s is a pattern occurring in $\pi_1[i_1 :]$ by the subsequence s_1 and $\min(s_1) = \min_1$ and $\max(s_1) = \max_1$ and s is occurring in $\pi_2[i_2 :]$ by the subsequence s_2 and $\min(s_2) = \min_2$ and $\max(s_2) = \max_2$ $\}$

We show that this family of problems are closed under induction.

BASE :

$$\mathrm{LCS}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(|\pi_1|, |\pi_2|) = \begin{cases} 1 & \text{if } \min_1 = \pi_1[j] = \max_1 \\ & \quad \text{and } \min_2 = \pi_2[j] = \max_2 \\ 0 & \textit{otherwise} \end{cases}$$

STEP :

$$\mathrm{LCS}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1,i_2) = max \begin{cases} \mathrm{LCS}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1,i_2+1) \\ \\ \mathrm{LCS}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1+1,i_2) \\ \\ \mathrm{M}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1,i_2) \end{cases}$$

with

$$\mathrm{M}^{\pi_2,\min_2,\max_2}_{\pi_1,\min_1,\max_1}(i_1,i_2) = \begin{cases} 1+\mathrm{LCS}^{\pi_2,\pi_2[i_2],\max_2}_{\pi_1,\pi_1[i_1],\max_1}(i_1,i_2+1) & \pi_1[i_1] < \min_1 \\ & \text{and } \pi_2[i_2] < \min_2 \\ \\ 1+\mathrm{LCS}^{\pi_2,\min_2,\pi_2[i_2]}_{\pi_1,\min_1,\pi_1[i_1]}(i_1+1,i_2) & \pi_1[i_1] > \max_1 \\ & \text{and } \pi_2[i_2] > \max_2 \\ \\ 0 & \text{otherwise} \end{cases}$$

For every pair i, j we either ignore the element of π_1, either ignore the element of π_2, either we match as the same step (if possible). Those relation lead to a $O(|\pi_1|^3|\pi_2|^3)$ time and $O(|\pi_1|^3|\pi_2|^3)$ space algorithm.

8 Conclusion