Pattern matching for (231, 213)-avoiding permutations

Both Emerite Neou[1,3] Romeo Rizzi[2] and Stéphane Vialette[3]

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

Abstract. bla bla...

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). During the last decade, the study of the pattern containment on permutations has become a very active area of research.

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 [5]). The permutation containment problem is **NP**-hard [5]. It is, however, polynomial-time solvable by brute-force enumeration if σ has bounded size. Improvements to this algorithm were presented in [2] and [1], the latter describing a $O(|\pi|^{0.47k+o(|\sigma|)})$ time algorithm. Bruner and Lackner [6] 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.) Of particular importance, it was recently proved that the permutation containment problem is fiexed-parameter tractable for parameter $|\sigma|$ [8].

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 [7], 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 [2].

Algorithmic issues for 321-avoiding patterns containment has been investigated in [9]. The pattern containment problem is also solvable in polynomial-time for separable patterns [10] (see also [5]). 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 [3], 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 [11] 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 [6] proved that deciding whether a vincular pattern σ of length k occurs in a permutation $\pi \in \mathfrak{S}_n$ 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 231 and 213 is $t_0 = 1$ for $n = 0$ and $t_n = 2^{n1}$ for $n \geq 1$ [13]. The permutations that do not contain the permutation pattern 231 are exactly the *stack-sortable permutations* and they are counted by the Catalan numbers [12]. 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.

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 $12 \ldots n$. 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 $\pi[i : n]$. 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 $\pi = 3\mathbf{91867}4\mathbf{5}2$ (or $\pi = 3\mathbf{91867}\mathbf{45}2$ or $\pi = 3\mathbf{91867}45\mathbf{2}$). 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 $\pi = \pi_1\pi_2 \dots \pi_n$ avoids 321, i.e., has no decreasing subsequence of length 3, then its reverse, $\pi' = \pi_n\pi_{n-1} \dots \pi_1$ 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 $\pi = \pi_1\pi_2 \dots \pi_n$, 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 $\mathrm{s}_\pi(1)\,\mathrm{s}_\pi(2)\dots\mathrm{s}_\pi(n-1) \in \{a, d\}^{n-1}$ 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 $\pi = 12\,11\,10\,1\,2\,3, 4\,9\,8\,5\,6\,7$ is $\mathrm{s}_\pi = dddaaaaddaa$. 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 $12 \dots k$ and the bottom row is a permutation $\sigma_1\sigma_2 \dots \sigma_k$). We have the following conditions on the top and bottom rows of σ:

- If the bottom line of σ contains $\underline{\sigma_i\sigma_{i+1} \dots \sigma_j}$ then the letters corresponding to $\sigma_i\sigma_{i+1} \dots \sigma_j$ 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 $\overline{i\,i+1 \dots j}$ then the letters corresponding to $\sigma_i, \sigma_{i+1}, \dots, \sigma_j$ 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 ${}^u lcorner 1$ 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.

A bivincular pattern avoids a set of permutations P if its bottom row $\sigma_1\sigma_2 \dots \sigma$ avoids every permutation in P.

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

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

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* $\pi[i:]$*.*

Corollary 2. *Let* $\pi \in \mathrm{Av}_n(213, 231)$*. We have*

$$\mathrm{s}_\pi(i) = \begin{cases} a & \text{if } \pi[i] \text{ is the minimal element of } \pi[i:] \\ d & \text{if } \pi[i] \text{ is the maximal element of } \pi[i:] \end{cases}$$

Lemma 2. *Given two permutations* π*,*σ *avoiding (231,213), If* s_σ *is a motif of stripe* s_π *by a subsequence* s*, then* σ *is a motif of* π *by subsequence* s*.*

Proof. We prove the lemma by induction on the size of the motif. Suppose that we have the motif (12), thus the stripe is a. If there is a match in stripe, then there exists two (contiguous) increasing elements in the text, which is a match in permutation. Same goes for the permutation (21). Now, suppose that the lemma is true for every motif of size n. Let σ be a motif of size $n+1$ avoiding $(213, 231)$, and π a permutation avoiding $(213, 231)$. If there is a match of s_σ in s_π by subsequence s, then there is a match of $\mathrm{s}_\sigma[2:]$ in $\mathrm{s}_\pi[2:]$ by subsequence $s[2:]$, and by induction there is a match of $\sigma[2:]$ in $\pi[2:]$ by subsequence $s[2:]$. Moreover suppose that the first element the stripe of the motif is ascend (descent), and by lemma 1 the minimal (maximal) element, and remark that a subsequence of a permutation avoiding (231,213) is also avoiding (231,213), thus the first element of the subsequence s is either the minimal (maximal) element which complete our proof.

Proposition 1. *Let* $\sigma \in \mathrm{Av}_k(213, 231)$ *and* $\pi \in \mathrm{Av}_k(213, 231)$*.* σ *is a motif of* π *if and only if* s_σ *is a subsequence of* s_π*.*

Proof. $\Rightarrow$ Suppose that σ is a motif of π by subsequence s. By definition s normalized is equal to σ so they have the same stripe.
$\Leftarrow$ This is proved by lemma 2.

Proposition 1 Let σ and π be two permutations avoiding (231,213), we can decide in linear time if σ appear in π.

Proof. To solve the permutation pattern for a permutation in $\mathrm{Av}_n(213, 231)$, we find a find a match of the stripe of the pattern in the stripe of the text. This can be done linearly by a greedy algorithm, by matching a step whenever it is possible. And thank to the definition of the definition of a stripe in **Corollary 1**, we do not have to compute the stripe advance. Therefore we have an linear (online) algorithm.

4 Pattern Matching With Motif Avoiding (231,213)

This section focus on the pattern matching problem when only the motif avoid $(231, 213)$.

Proposition 2. *Let σ, π be two permutation, with σ a permutation avoiding (231,213), we can decide in $O(|\sigma||\pi|^2)$ time and $O(|\sigma||\pi|^2)$ space if σ is a motif of π.*

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. 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 $\mathrm{F}(i-1)\cdots\mathrm{F}(1)$, 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 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 $\mathrm{F}(i+1)\,\mathrm{F}(i)$, $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$, every elements 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 that the maximal (minimal) element of $\mathrm{F}(i)\,\mathrm{F}(i-1)\cdots\mathrm{F}(1)$ is $\mathrm{L}(i-1)$.

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 :

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

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:

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

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

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 [4]).

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 to the maximum element of the rest of the match ie inferior 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 Pattern Matching With Bivincular Motif Avoiding (231,213)

This section gives an algorithm to detect bivincular permutation in polynomial time in the case where the motif avoid $(231, 213)$.

Proposition 3. *Let (σ, X, Y) be a bivincular motif avoiding (231,213), π a permutation, we can decide in $O(|\sigma||\pi|^3)$ time and $O(|\pi|^3)$ space if σ appear in π.*

Proof. We consider the following problem :

Given a bivincular motif (σ, X, Y) avoiding $(231, 213)$, a text π, i,j , $i < j$.

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(i,j) = \begin{cases} true & \text{If } \sigma[i:] \text{ is a motif of } \pi[j:] \\ & \text{with every element in} \\ & [\min,\max] \\ false\ otherwise \end{cases}$$

$\mathbb{P}_{\sigma,\pi,\min,\max}$ is closed under induction. it can be solved by means of the following relations:

BASE :

if $|\sigma| \notin X$:

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|,j) = \begin{cases} true & \text{if } \min < \pi[j] < \max \\ false\ otherwise \end{cases}$$
$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|+1,j) = true$$

if $|\sigma| \in X$:

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|,|\pi|) = \begin{cases} true & \text{if } \min < \pi[|\pi|] < \max \\ false\ otherwise \end{cases}$$
$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|+1,|\pi|) = true$$

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

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|,j) = \begin{cases} true & \text{if } \min < \pi[j] < \max \\ & \pi[j] \text{ is the maximal element} \\ false\ otherwise \end{cases}$$
$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|+1,j) = true$$

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

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|,|\pi|) = \begin{cases} true & \text{if } \min < \pi[|\pi|] < \max \\ & \pi[|\pi|] \text{ is the maximal element} \\ false\ otherwise \end{cases}$$
$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(|\sigma|+1,|\pi|) = true$$

STEP :

$$\mathbb{P}_{(\sigma,X,Y),\pi,\min,\max}(i,j) =$$

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

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 $\pi[j+1] = \sigma[i]-1$.

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 $\pi[j+1] = \sigma[i]+1$.

There are two cases to consider searching for a match :
If $|\sigma| \in X$ there exists a match if and only if
$\bigcup_{0<\min<\max<|\pi|} \mathbb{P}_{\sigma,\pi,\min,\max}(0,0)$ is true.

If $|\sigma| \notin X$ there exists a match if and only if
$\bigcup_{0<k<|\pi|,0<\min<\max<|\pi|} \mathbb{P}_{\sigma,\pi,\min,\max}(0,k)$ is true.

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 case for $M(\pi)$ follows the same idea.

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 4 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 [4]), 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 solve in $O(|\pi_1|^3|\pi_2|^3)$.*

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.

$S_{\pi_1,\pi_2}(\min_1, \max_1, \min_2, \min_2, i_1, i_2) = \max\{\ |m|\ |$ 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 :

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

STEP :

$$S_{\pi_1,\pi_2,\min_1,\max_1,\min_2,\max_2}(i_1, i_2) =$$
$$\max \begin{cases} S_{\pi_1,\pi_2,\min_1,\max_1,\min_2,\max_2}(i_1, i_2 + 1) \\ S_{\pi_1,\pi_2,\min_1,\max_1,\min_2,\max_2\,2}(i_1 + 1, i_2) \\ S'_{\pi_1,\pi_2,\min_1,\max_1,\min_2,\max_2}(i_1, i_2) \end{cases}$$

with $S'_{\pi_1,\pi_2,\min_1,\max_1,\min_2,\max_2}(i_1,i_2) =$

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

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

References

1. S. Ahal and Y. Rabinovich. On Complexity of the Subpattern Problem. 22(2):629–649, 2008.
2. M.H. Albert, R.E.L. Aldred, M.D. Atkinson, and D.A. Holton. Algorithms for pattern involvement in permutations. In *Proc. International Symposium on Algorithms and Computation (ISAAC)*, volume 2223 of *Lecture Notes in Computer Science*, pages 355–366, 2001.
3. D. Avis and M. Newborn. On pop-stacks in series. *Utilitas Mathematica*, 19:129140, 1981.
4. Sergei Bespamyatnikh and Michael Segal. Enumerating longest increasing subsequences and patience sorting, 2000.
5. P. Bose, J.F.Buss, and A. Lubiw. Pattern matching for permutations. *Information Processing Letters*, 65(5):277–283, 1998.
6. M.-L. Bruner and M.Lackner. A fast algorithm for permutation pattern matching based on alternating runs. In F.V. Fomin and P. Kaski, editors, *13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT), Helsinki, Finland*, pages 261–270. Springer, 2012.
7. M. Crochemore and E. Porat. Fast computation of a longest increasing subsequence and application. *Information and Compututation*, 208(9):1054–1059, 2010.
8. S. Guillemot and D. Marx. Finding small patterns in permutations in linear time. In C. Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Portland, Oregon, USA*, pages 82–101. SIAM, 2014.
9. S. Guillemot and S. Vialette. Pattern matching for 321-avoiding permutations. In Y. Dong, D.-Z. Du, and O. Ibarra, editors, *Proc. 20-th International Symposium on Algorithms and Computation (ISAAC), Hawaii, USA*, volume 5878 of *LNCS*, page 10641073. Springer, 2009.
10. L. Ibarra. Finding pattern matchings for permutations. *Information Processing Letters*, 61(6):293–295, 1997.
11. S. Kitaev. *Patterns in Permutations and Words*. Springer-Verlag, 2013.
12. Donald E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1997.

13. R. Simion and F.W.Schmidt. Restricted permutations. *European Journal of Combinatorics*, 6(4):383406, 1985.