Guillotine Cuts & Pattern Avoidance in Permutations

Student Name: Siddhartha Jain
Roll Number: 2016269

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Applied Mathematics

on April 15, 2019

BTP Track: Research

BTP Advisors
Dr Rajiv Raman
Dr Samrith Ram

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **"Guillotine Cuts & Pattern Avoidance in Permutations"** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Applied Mathematics* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr Rajiv Raman** & **Dr Samrith Ram**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

....
Siddhartha Jain

Place & Date: ....

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

....
Dr Rajiv Raman

Place & Date: ....

Abstract

Given a set of axis-parallel rectangles on a plane, can you cut out a constant fraction of them using only Guillotine cuts? We try to answer this question in the project. The general problem of cutting out convex sets was investigated by Pach & Tardos [11]. But there has been a lot of interest in proving better bounds for rectangles. This is because if the answer to the aforementioned question is in the affirmative, it automatically gives us a polytime $O(1)$-approximation to MISR, as stressed in a paper by Abed, Chalermsook et al. [1], which made progress by showing a bound of $n/81$ for the special case of squares.

In our investigation of these structures a strong connection with permutations showed up. This was mainly due to a combinatorial bijection between Floorplans and Baxter permutations given by Ackerman et al. [2]. Combined with the work of Young et al. [13], it inspired a strategy to prove bounds which works for all rectangle sets: delete just enough rectangles to delete all **pinwheels**, which we define.

Moreover, previously unexplored (seemingly) yet natural questions about the field of Pattern Matching / Avoidance in Permutations presented themselves in our study. We detail numerous results and conjectures in this area.

Keywords: Combinatorial Geometry, Geometry, Rectangles, Permutations, Pattern Matching, Guillotine cuts, Independent Sets

Acknowledgments

I am deeply grateful to Prof Rajiv Raman for introducing me to this problem and giving me the opportunity to work on it. I would like to thank Hanit Banga from IIIT Delhi and Anh Mai from NYU AD for numerous fruitful discussions. I would also like to thank Siddharth Yadav and Muhammad Falak Reyaz Wani from IIIT Delhi for helping me learn and implement a parallel program in Go. IIIT Delhi has also provided us with the computing power necessary to carry out this project.

Work Distribution

Chapter	Task	Completion Date
3.1	Squarability	May 15 2018
2	Literature review for scheme	Jun 10 2018
2	Learning basics of Go	Jun 12 2018
2	Implementation	July 6 2018
5.1	Floorplans	July 2 2018
4.2	Gap example	Aug 15 2018
3	Rectangles: Observations	Feb 2019
5	Rectangles: Negative Results	March 2019
6	Pattern Avoidance: Algorithmic Results	Dec 2018
7	Pattern Avoidance: Combinatorial Results	March 2019

Contents

Chapter 1

Introduction

The physical intuition for Guillotine cuts is provided very well by Abed, Chalermsook et al [1] "Imagine a wooden plate with a set of non-overlapping geometric objects painted on it. How many of them can a carpenter cut out using a panel saw making guillotine cuts, i.e., only moving forward through the material along a straight line until it is split into two pieces?"

For the case of axis-parallel rectangles, we have a conjecture which follows.

Conjecture 1.1. *For any set of axis-parallel rectangles on a plane, the optimal Guillotine cutting sequence saves at least* $\frac{n}{2}$ *rectangles.*

There are many examples, well quoted in literature, where you must kill $n/2$ rectangles. Therefore we can't do better.

What is interesting about all such examples is that they exploit the structure of pinwheels. Before we define what those are, we define an equivalence relation on sets of rectangles such that those belonging to the same class are combinatorially equivalent.

Definition 1.2. *Consider two rectangle sets* R_1, R_2*. Let the interval order defined by the projections on the* X*-axis be* X_1, X_2 *and* Y*-axis be* Y_1, Y_2*.*

Then R_1 *and* R_2 *are said to have the same* spatial relations *iff there is some permutation* π *such that* $\pi(X_1) = X_2$ *and* $\pi(Y_1) = Y_2$*.*

So we consider all sets with the same spatial relations the same. Now, we define a pinwheel.

Definition 1.3. *A* pinwheel *is a rectangle set such that* $|R| = 4$*, and there is no cut with at least one rectangle on either partition which kills zero rectangles.*

Lemma 1.4. *There is only one pinwheel upto labelling of spatial relations.*

Initially, our focus for the project was Enumeration, which was done in order to try and disprove Conjecture 1.1 through counterexamples. You can take a look at all the code written for this project at it's GitHub repository. Ultimately, we could not disprove this conjecture experimentally and still believe it is true.

After reading recent work in this area and the related area of Pattern Matching we were able to contribute some results and motivate a new set of problems in Pattern Matching, as mentioned in this thesis.

Chapter 2

Enumeration

Since there has not been progress for rectangles ever since Pach and Tardos proved a bound of $\Omega(\frac{n}{logn})$ [11], we decided what the problem needed was data through computer search. The enumeration scheme has 3 phases, as explained in the first section.

2.1 Scheme

2.1.1 Baxter Permutations

The first phase of the Enumeration scheme calls for enumerating Baxter permutations. Baxter permutations are those which avoid the vincular patterns $2\underline{41}3$ and $3\underline{14}2$. Fortunately, there was work done by Bouvel et al [5], which used a Generating Tree scheme to enumerate.

2.1.2 Mosaic Floorplans

The second phase consists of interpreting the permutations as floorplans. This is acheived through a bijection given by Ackerman [2], who gives an efficient $O(n)$ algorithm to convert any Baxter permutation to the corresponding Mosaic floorplan. Moreover, this bijection is preserved between Seperable permutations and Guillotine/Slicing floorplans. Therefore, with the help of work done by Bose et al [4], who give a linear time algorithm to recognise Seperable permutations, we were able to cut down run-time, since in all Slicing floorplans we can trivially save all rectangles.

2.1.3 Optimal Guillotine Cutting Sequence

This is computed for an input set using a dynamic programming algorithm which stores the corresponding seq, k pair for any subregion of the input. Here seq is the sequence of cuts in the optimal fashion and k is the minimum number of rectangles killed while separating all other rectangles.

2.2 Exploiting Symmetry

Since the Guillotine cut property is unchanged on the Dihedral group (D_4) action, we can use this to reduce the number of instances for which we need to compute the Optimal Cutting Sequence by a factor of 8. Moreover, we do not even need to construct floorplans and carry out the expensive process of performing reflections/rotations and checking equivalence. The corresponding Baxter permutations can be converted among one another by three operations: inversion, complement & reversal.

2.3 Implementation

The code is written in *Go*. Advantages of using Go:

- Fast, inbuilt concurrency tools
- Designed to be easy to maintain (in case of changes/optimisations)

Moreover, other higher-level languages like Python, Sage did not offer libraries to reduce implementation size. They would also be slower than Go. Hence, we made the decision to use Go for our project.

In hindsight, using Julia would have been better for the project since it would give similar speed and ease of running in parallel, and it has been steadily gaining traction in academia.

2.4 Remarks

The enumeration scheme only considers Floorplans and not general sets. While this is an obvious drawback, it's a necessary limitation. The number of Baxter permutations (and therefore Mosaic Floorplans) is well understood, and is asymptotically $\theta(\frac{8^n}{n^4})$. You can also see the first few terms on OEIS [10], sequence A001181. On the other hand, the number of rectangle sets which are combinatorially distinct is not well understood. Yet, we have a lower bound proved by Silvanus, Vygen [12], which is $\Omega(n!\frac{c^n}{n^4})$ where $c = 4 + 2\sqrt{2}$. As you can see, this is already unfeasible for any reasonably big n.

We also think it's worth investigating Mosaic floorplans because they are well understood. Since Mosaic/Slicing floorplans correspond exactly to Baxter/Seperable permutations, one way of proving a bound on the number of rectangles one can save in Mosaic Floorplans is to prove a bound on the size of the largest Seperable permutation induced by a Baxter permutation.

Hence the question becomes, if it is true that we can save $\Omega(n)$ rectangles in Mosaic floorplans, does that imply we can save $\Omega(n)$ rectangles in any set? We answer this question negatively in

Section 5.1.

2.5 Results

Find below a table documenting the results of the experiment. The columns are as follows.

- Size of set: the number of rectangles in the floorplan.
- Rectangles killed: the number of rectangles killed by the best cut sequence.
- Count of extremal: the number of floorplans where the best cut sequence kills the worst number, listed in previous column.

A document containing a list of all the extremal examples can be found on the GitHub repository, with the name FullResults.txt.

Extremal Examples		
Size of set	Rectangles killed	Count of extremal
5	1	1
6	1	14
7	1	134
8	2	1
9	2	85
10	2	1900
11	3	14
12	3	1140

Chapter 3

Rectangles: Observations

3.1 Characterising Squarable Sets

Definition 3.1. *We define a set of rectangles on a plane to be* squarable $\iff \exists$ *a set of squares which shares the same* spatial relations.

This is of interest since we already know a lower bound of $\frac{n}{81}$ for squares [1]. We demonstrate a family of sets we call C_n, which are provably non-squarable. Here $n = (n_1, n_2, n_3, n_4)$ is a tuple of size 4. Notationally, on writing C_k where k is a single integer, k represents the tuple (k, k, k, k). Now, C_n corresponds to a classic pinwheel shape where instead of one rectangle each edge of the pinwheel is composed of a *block* of n_i parallel rectangles. For a visual representation refer to Fig 3.1.

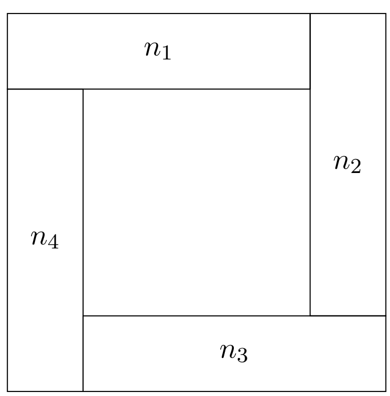

Figure 3.1: C_n, n_i represents as many parallel rectangles stacked

Lemma 3.2. C_n *is non-squarable whenever* $n_i \geq 3$ $\forall i \in \{1, 2, 3, 4\}$.

Proof. We assume C_3 is squarable and produce a contradiction. The proof for C_n follows similarly. Now, we use $r_{k,l}$ to represent the l^{th} rectangle of the k^{th} block.

Consider the representation of C_3 using only squares. Let the horizontal edges of $r_1, 2$ be segments on $y = c_1, y = c_2$. Now, for some $c \in (c_1, c2)$ and $\epsilon > 0$, let us define two lines

$l_1 \equiv y = c_1 + \epsilon$ and $l_2 \equiv y = c_2 - \epsilon$. Also, let the side of $r_{1,2}$ be s_1. Since l_1, l_2 must intersect all rectangles in block 2, if s_2 is side of block $r_{2,2}$ then $s_2 \geq s_1 + \epsilon$. Repeating this argument, if s_3 is the side of block $r_{3,2}$ then $s_3 \geq s_2 + \epsilon$. Repeat again for $r_{4,2}$ to get $s_4 \geq s_3 + \epsilon$. But we can repeat yet again to get

$$s_1 \geq s_4 + \epsilon \Rightarrow s_1 \geq s_1 + 4\epsilon \Rightarrow \epsilon \leq 0 \Rightarrow\Leftarrow$$

□

Conjecture 3.3. *All non-squarable sets induce a member of C_n as a subset.*

3.2 Min cut kills $\leq \frac{n}{4}$

Note that if we allow arbitrary lines then trivially there is a cut which kills 0 rectangles. By a cut we mean a line that has at least one rectangle in each halfplane.

Theorem 3.4. *There is a cut which kills $\leq \frac{n}{4}+1$ rectangles on any disjoint axis parallel rectangle set.*

Proof. Note that it is enough to look at all the lines which contain edges of rectangles. Now, since we have a finite set, taking all the lines parallel to the X-axis we have a lowermost cut and an uppermost cut. Similarly we have a leftmost and rightmost. Let us call these the extremal cuts. One possibility is that one of these kills $\leq \frac{n}{4} + 1$ rectangles. Then we're done.

Assume this is not true. Illustrated in Fig 3.2. Let R_l, R_r, R_u, R_b be the set of rectangles cut by the leftmost, rightmost, uppermost and lowermost rectangle respectively. If all of these are disjoint we have arrived at a contradiction. But this is not true in general, parallel lines can have an intersection. Intersecting lines cannot have more than one common rectangle because it would imply intersecting rectangles.

But take a rectangle $r \in R_l \cap R_r$, since we have > 1 rectangles either its top or bottom edge is not contained in either R_u or R_b. Take the line containing this edge. It intersects at most 2 rectangles, since r must span almost the entire width of this set. Hence we have produced a cut killing 2 if there is an intersection, and if such an intersection does not exist, since all 4 lines kill $> \frac{n}{4} + 1$ rectangles and we counted atmost 4 rectangles twice, we have demonstrated more rectangles than we started with. $\Rightarrow\Leftarrow$ □

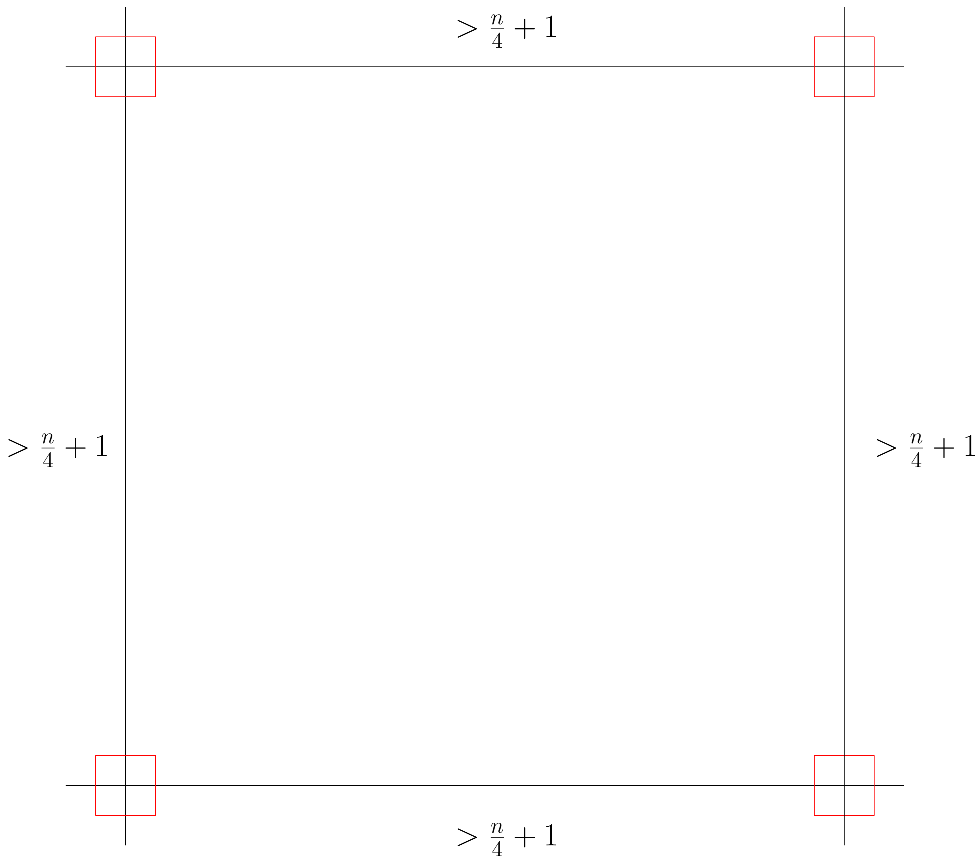

Figure 3.2: Assuming they all kill $> \frac{n}{4} + 1$ leads to a contradiction

Since one can easily construct an example where you must kill $\frac{n}{4}$ rectangles, this bound is tight.

3.3 Save $\leq \frac{n}{2}$ if same height

In this section we will consider the situation where all the rectangles have the same length along one axis, say k. Assume without loss of generality their heights are same. Imagine drawing infinite parallel horizontal lines at a uniform distance of $2k - \epsilon$ for some small $\epsilon > 0$. Now, note that every "strip" contains non stacked rectangles, which can be separated. But we don't know how many rectangles these lines kill. Let us do a random shift in the range $[0, 2k - \epsilon]$. The expected fraction of rectangles killed is

$$\lim_{\epsilon \to 0} \frac{k}{2k - \epsilon} = \frac{1}{2}$$

Hence, there exists some shift such that we can save as many rectangles.

Chapter 4

Relation with Permutation Patterns

4.1 Motivation

We have a bijection between Mosaic Floorplans and Baxter Permutations [2]. This bijection is preserved between Separable Permutations and Slicing Floorplans. All slicing floorplans are Guillotine separable, and this is true for induced permutations. Our hope was that the other direction is also true and we have a bijection. As we will show, this is not true.

Remarkably, this means that to show a linear lower bound on rectangles we have can save in Floorplans it is enough to show a linear bound on the largest Separable permutation induced in any Baxter permutation, if it is true. These kinds of bounds are explored in Chapter 7.

What is also interesting is that because of the work of Young et al. [13], we know that if a rectangle set does not contain a pinwheel, there is a Floorplan with the same spatial relations. This also means that to get a set that is Guillotine separable (in general, not restricted to Floorplans), it is enough to delete enough rectangles to get rid of all pinwheels! That is a novel approach to trying to prove this combinatorial bound.

4.2 Gap between induced permutation and Guillotine set

As demonstrated in Fig 4.1, without changing the permutation achieved using the mapping from [2], we go from an example where we must kill $\frac{n}{6}$ rectangles to 0. The marked rectangle indicates recursively substituting a floorplan. The dotted red line indicates the first cut we need to make to separate all rectangles.

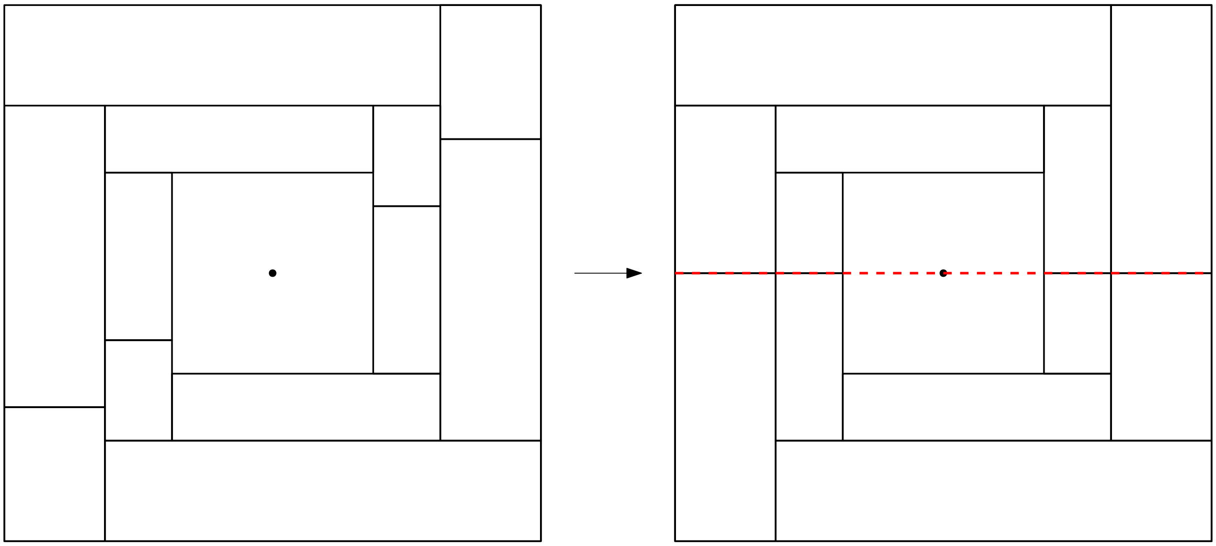

Figure 4.1: We go from having to kill $\frac{n}{6}$ rectangles to *zero*.

4.3 Permutation Classes of Interest

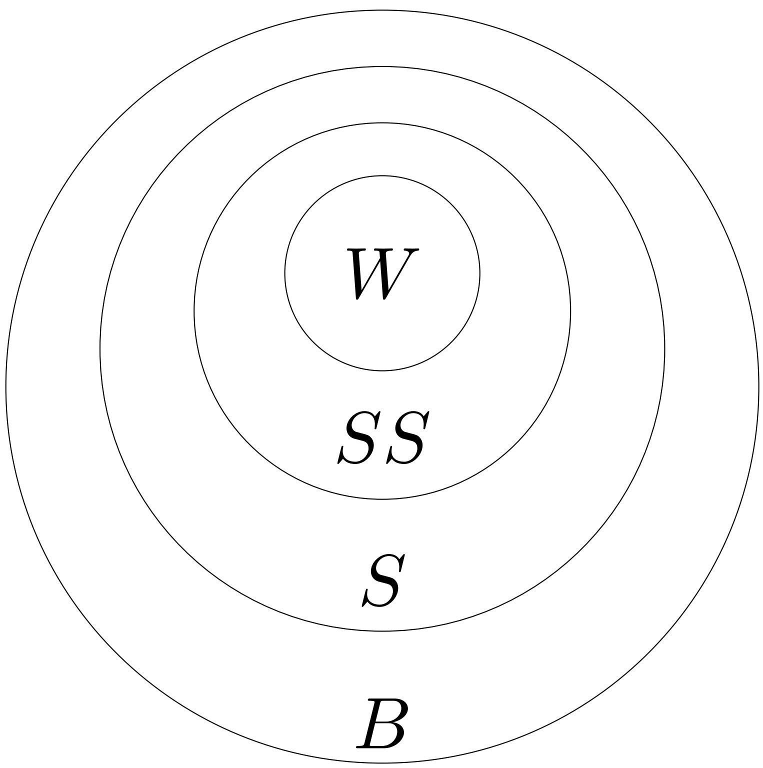

Figure 4.2: W = Wedge, SS = Stack Sortable, S = Separable, B = Baxter

Chapter 5

Rectangles: Negative Results

5.1 Minimal Extension of Rectangles Sets to Mosaic Floorplans

In this section we provide a negative result. First, we define some terms.

Definition 5.1. *The* Collapsed set *of a corresponding rectangle set is the geometric equivalent of compressing all the "empty space" between rectangles. First collapse with respect to the X axis and then Y axis, without changing spatial relations.*

Definition 5.2. Holes *in a Collapsed set are the nuber of rectangles required to complete into a Mosaic Floorplan.*

Definition 5.3. Charge *of a rectangle in a Collapsed set is the net number of Holes that can be attributed to it. Each hole has to dissipate 1 Charge among all the rectangles that share a boundary with it.*

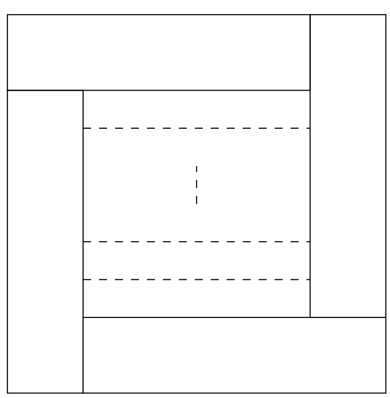

Figure 5.1: Pinwheel; horizontal dashed lines represent thin rectangles

Lemma 5.4. *A lower bound on the maximum average Charge in a Collapsed set is 1.*

Motivation Let the maximum average Charge be c. Let the fraction of rectangles we can save in Mosaic floorplans be $\frac{1}{s}$. If $1 + c < s$, we can save $\Omega(n)$ rectangles in all sets.

Proof. Referring to Fig 5.1, we start with a classic pinwheel and we put k parallel rectangles inside.
Now you can see that a pinwheel has 1 hole. If we add a rectangle inside, the number of holes increases to 2. Similarly, with k rectangles we have $k+1$ holes. The total number of rectangles is $k+4$. Since $\frac{k+1}{k+4}$ converges to 1, we have demonstrated that $\exists$ a rectangle set with average charge $c \in (0,1)$ where c is arbritarily close to 1. Hence, proved. □

Now, we know $s \leq 2$. Therefore what we have shown here is that this approach cannot be used to extend the result from Mosaic Floorplans to Rectangle sets.

5.2 Separable permutations for general sets

We can define a surjection from rectangle sets to permutations. We do this by defining two linear orders, and permuting one with respect to the other. To define the orders, just take the dictionary ordering on the intervals which are the horizontal or respectively vertical projections of the rectangles.

Since a pinwheel induces 2413 or 3142 as a pattern, a the preimage of a Separable permutation is a set of rectangle sets where we can save all rectangles. Hence this seems like a promising approach after the negative result of the previous section. But turns out this does not help.

Theorem 5.5. *In the $n \times n$ grid permutation [7], the largest induced Separable permutation is of size $O(n)$.*

Proof. Notice that if we take any 4 points from different rows and columns, they induce 2413 or 3142. Hence we can take 2 rows, 2 columns or a row and a column. In all cases, we get $\leq 2n$ points. □

Chapter 6

Pattern Avoidance: Algorithmic Results

We are primarily interested in the following problem.

Definition 6.1. *Let $P = \{\rho_1, \rho_2 \ldots \rho_r\}$ be some set of "patterns" and π some "text," both of which are permutations for our purposes. Then define* LongestPatternAvoiding*(P, π) as the length of the largest permutation induced in π that avoids all the patterns in the set P.*

This can be seen as a generalisation of the PatternMatching problem, which is well studied in literature. This immediately implies the problem in NP-hard, as explained in Lemma 6.2. We believe that it is not FPT either. Refer to Section 6.1 for more details.

Below you can find a table with a summary of all the results in the field. Since we could not find any prior literature on this subject we also include results by other authors that give an algorithm for a problem in this area as a review.

Time complexity of LargestPatternAvoiding(ρ, π)		
Name	**Pattern**	**Upper bound**
Monotone	12 21	$O(n \log\log n)$ [3]
k-monotone	$\rho = 12 \dots k$ $\rho = k \dots 21$	$O(n^{k-1} \log\log n)$
Stack Sortable	213 312 231 132	$O(n^6)$ $O(n^6)$
Wedge-like	$231, 213$ $132, 312$	$O(n \log n)$ [9]
Unimodal	$231, 132$ $213, 312$	$O(n^2 \log\log n)$
Pseudo-monotone	$231, 312$ $132, 213$	$O(n^2 \log\log n)$
Separable	2413, 3142	$O(n^6)$
(Arbitrary)	$\|\rho\| = k$	$O(2^{n+k^2} n)$ [7] [6] $O(2^n n^{0.44k+o(k)})$ [8] $O(2.415^n)$ [8]

6.1 Hardness

Lemma 6.2. *When ρ is not fixed, the problem of finding largest induced member of $Av(\rho)$ in given π, that is* LargestPatternAvoiding*(ρ, π) is* NP*-Hard.*

Proof. This follows from the fact that PatternMatching is NP-Hard. The following two statments are equivalent:

- $\rho \not\leq \pi$
- $\pi \in Av(\rho)$

Hence $\rho \not\leq \pi$ iff answer to LargestPatternAvoiding is n. □

Lemma 6.3. *If ρ is monotone, $|\rho| = k$, then* LargestPatternAvoiding *can be solved in $O(n^{k-1} \log \log n)$ time.*

Proof. We guess $k-2$ "break points" to divide our array into $k-1$ parts. In $O(n \log \log n)$ time we compute LIS/LDS in each part and combine the answer. By doing so, we introduce $\leq k-1$ ascents/descents. Hence our algorithm is correct. And since any $Av(\rho)$ permutation can be written as these $k-1$ factors, it is optimal.

This algorithm runs in $O(n^{k-2} n \log \log n) = O(n^{k-1} \log \log n)$ time. □

All our attempts to construct a polytime algorithm for any *non-monotone* π such that $Av(\pi) \not\subseteq S$ have failed. We have polytime algorithms for all interesting $Av(\pi)$ and $Av(\pi_1, \pi_2)$ which are a subset of Separable permutations. Finally, when we try to make general algorithms for finding largest induced member of $Av(\rho)$, the main hurdle to getting a $O(n^{O(k)})$ algorithm is related combining subproblems. This issue goes away with Separable permutations.

Conjecture 6.4. *For any set $P = \{\pi_1, \pi_2 \ldots \pi_k\}$ such that for some i, π_i is not monotone, if $Av(P) \not\subseteq S$ then finding the largest induced member of $Av(P)$ is $NP-hard$.*

6.2 Separable

We claim the we can find the largest induced Separable permutation in any given input permutation of size n in $O(n^6)$ time. Let $LS(i, j, s, L, \odot)$ be the largest separable permutation between indices i, j with values between s, L where $i < j$ and $s < L$ where the first step of the decomposition is $\odot = \oplus, \ominus$. Now since the permutation is Separable, we can choose a point on the rectangle of the plot so that the left (right) lies completely below the right (left) if the symbol is $\oplus(\ominus)$. Hence we can write the following recurrence.

$$LS(i, j, s, L, \oplus) = \max_{(p,k,\odot,\odot)} LS(i, p, s, k, \odot) + LS(p, j, k, L, \odot)$$

$$LS(i, j, s, L, \ominus) = \max_{(p,k,\odot,\odot)} LS(i, p, k, L, \odot) + LS(p, j, s, k, \odot)$$

Since we are taking max over $O(n^2)$ quantities and there are $O(n^4)$ subproblems, using memoization we can compute $max_{\odot=\oplus,\ominus} LS(0, n-1, 1, n, \odot)$ in $O(n^6)$ time.

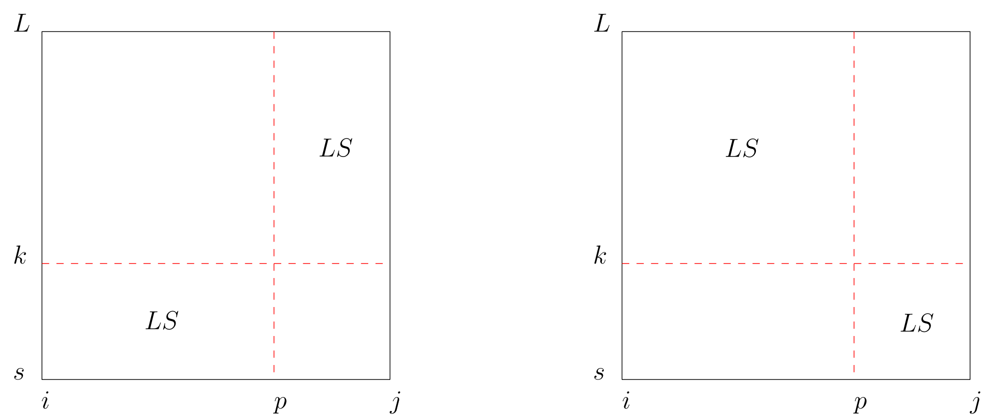

Figure 6.1: Two possibilities fixing p and k

6.3 Stack Sortable

To design an algorithm for this task, first we make an observation about the structure of a Stack Sortable permutation.

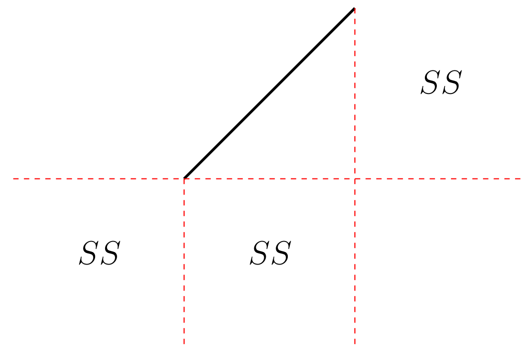

Figure 6.2: On the plot of our permutation, a SS permutation induces this struture

Here the X-axis corresponds to indices of the permutation and the Y-axis corresponds to the value. Figure 6.2 immediately suggests a recurrence to find the largest SS permutation.

$$LSS([i,j],[s,L])$$

is exactly

$$\underset{\substack{|\pi| \\ i \leq i' \leq j' \leq j \\ A[i'] \leq A[j']}}{\arg\max}\, \pi = LSS([i,i'],[s,A[i']]) \cdot A[i'] \cdot LSS([i',j'],[s,A[i']]) \cdot A[j'] \cdot LSS([j',j],[A[i'],L]) \quad (\star)$$

Here we want to find $LSS([0,n-1],[1,n])$. If $|j-i| \leq 3$ we can simply check if $A[i:j]$ matches 231 and reject if so. Using the recurrence, and memoization of LSS() values, we can compute the largest SS permutation in $O(n^6)$ time and $O(n^4)$ space.

Theorem 6.5. *The recurrence ($\star$) for LSS is correct.*

Proof. We proceed by PMI on $|j-i|$. We will use $\pi_{\leq i}$ to mean the word for π upto the element whose index is $\leq i$.

Base case: If $|j-i| \leq 3$, trivial.

Induction Hypothesis: For all $|j-i| < k$, LSS recurrence holds.

Now given $LSS([i,j],[s,L])$ such that $|j-i| = k$, let us denote $A[i:j] = \pi$ and the largest SS permutation in $A[i:j]$ by ρ. If $|\rho| \leq 3$ we are done.

Let j_0 be the index of the largest element in ρ. Now let i_0 be the index of the largest element in $\rho_{\leq j_0}$. We claim that $[i_0, j_0]$ maximises the value in the recurrence. Anything in ρ lying between the index i_0, j_0 must induce another SS permutation. By our choice of i_0, everything in this permutation must have a value $< A[i_0]$. Similarly for everything in ρ that lies before i_0. After j_0, again it must induce a SS permutation and by definition of how we chose j_0 everything must be $< A[j_0]$. Moreover if there was some element a in ρ after j_0 whose value was $< A[i_0]$, then $A[i_0]A[j_0]a$ induce a 231, which is not possible. Therefore, $[i_0, j_0]$ gives us ρ using our recurrence. ∎

Note that we can extend this algorithm to recognise the largest $Av(132)$ member since the pattern is simply the reverse. Moreover, for $Av(213)$ we simply need to reverse the inequalities in the recurrence and swap some indices, which further gives us an algorithm for $Av(312)$ by reversing.

Complexity for $Av(\pi)$	
Pattern	Upper bound
132	$O(n^6)$
213	$O(n^6)$
231	$O(n^6)$
312	$O(n^6)$
213, 231	$O(n \log n)$ [9]

6.4 Unimodal

For $Av(213, 312)$ notice that for any element, you cannot have elements which are larger than it on both the left and right. Because then one of the patterns will be induced. Hence a V can never be induced. Hence it is unimodal, and in fact it first increases then decreases.

So an algorithm to compute the largest member of $Av(213, 312)$ is to simply guess the min, split the permutation at that point. Now reverse the left half. Compute LIS for both halfs, reverse the left again and report their concatenation. We have n guesses for the min and we can compute LIS in each half in $O(n \log \log n)$ time. Hence the entire algorithm takes $O(n^2 \log \log n)$ time.

Similarly for $Av(231, 132)$ a Λ can never be induced. See Fig 6.3 for a visual representation of how the permutations look.

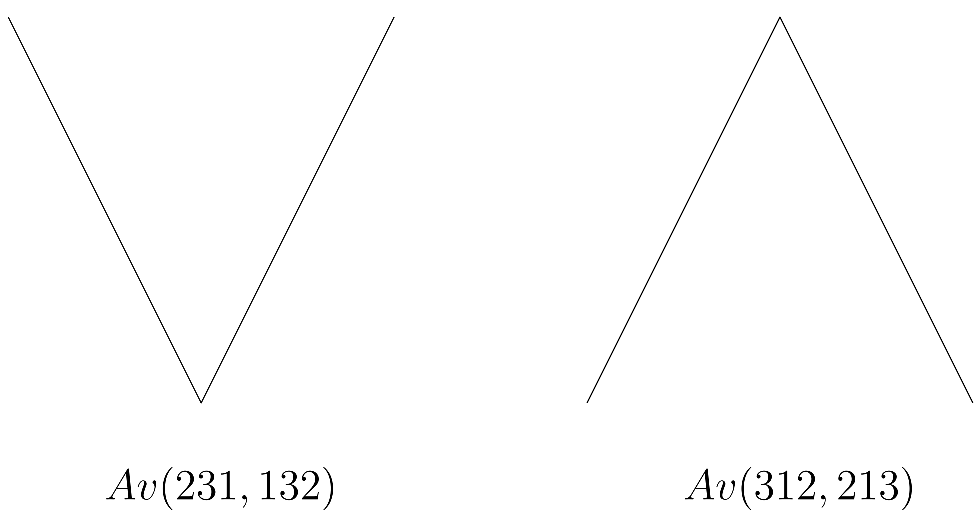

Figure 6.3: Both classes contain a specific kind of unimodal permutation

6.5 Pseudo-monotone

Consider $Av(231, 312)$. It has the structure in Fig 6.4. Hence inductively, it is composed of strictly decreasing factors each of which is stricly greater than the last. We solve this in $O(n^3)$ time in two ways. Then we improve to $O(n^2 \log n)$.

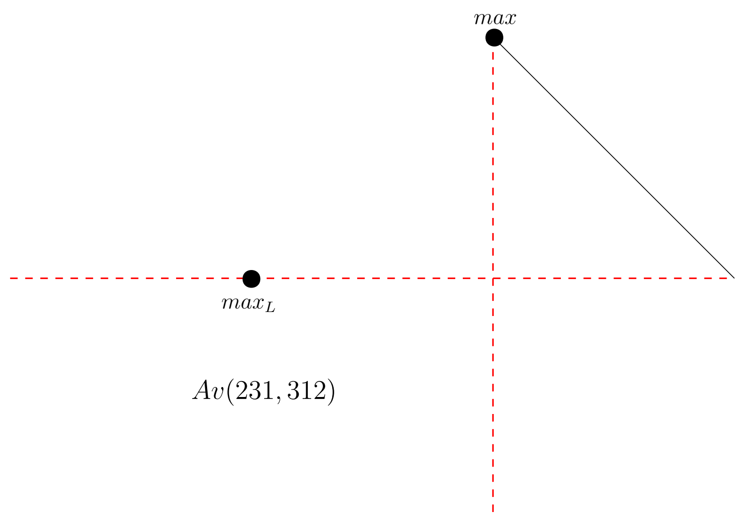

Figure 6.4: Structure imposed by avoidance of patterns

6.5.1 Approach 1

First in $O(n^2 \log n)$ time we compute the LDS ending at every index from every index before it. Define $TIME(n, k)$ as the time required to compute the longest pseudo-monotone permutation having k factors on a permutation with n elements.

Assuming we know how many factors the longest permutation has, say k, we can guess the min element of the middle one. Now for every element before we take the LDS from it ending at our element (might be empty) and recurse on both sides. Therefore we get:

$$TIME(n, k) \leq 2TIME(n, \frac{k}{2}) + c$$

$$TIME(n, k) \leq 2^{\log k} TIME(n, 1) + c(2^{\log k} - 1)$$

$$TIME(n, k) \in O(kn)$$

Define $TIME(n)$ as the max time required for a permutation of length n, Now our solution is max of results returned by all possible values of k, but $k \leq n$ hence we get:

$$TIME(n) = \sum_k TIME(n, k) = \sum_k O(kn) \in O(n^3)$$

6.5.2 Approach 2

Define $LDS(L, R, v)$ as the length of the LDS which starts at L (must include L), ends not after R, and all elements are $\geq v$. Then, either the element at R is in the permutation or not. Hence we get a recurrence.

$$LDS(L, R, v) = max(LDS(L, R-1, v), 1 + LDS(L, R-1, \pi(R)+1))$$

Let us also define $LPM(i, m)$ as the largest member of $Av(231, 312)$ in $\pi[: i]$ with the largest element *exactly* m. Also define $LPM_{MAX}(i, m) = \max_{x<m} LPM(i, x)$. Then if j is such that $\pi(j) = m$ then we can say the following.

$$LPM(i, m) = \max_{x \leq m} \; LPM_{MAX}(j-1, x) + LDS(j, i, x)$$

Using these recurrences we can solve $LPM_{MAX}(n, n+1)$ in $O(n^3)$ time, which is our solution.

6.5.3 Improvement of Approach 2

Define $LPM(i, m)$ as the answer among first i elements, such that the largest number used is *atmost* m. Let p be such that $\pi(p) = m$. And $LDS(L, R)$ is LDS which starts at L and ends at R, must include both indices. Then there are three (not mutually exclusive) possibilities:

1. We're not using the element at position i.
2. We're not using the element with value m.
3. We're using both position i and value m. Since m is max value used, and i is rightmost position possible, this forces the last chain to start at p and end at i.

Hence we get the following recurrence:

$$LPM(i, m) = max(LPM(i-1, m), LPM(i, m-1), LPM(p-1, \pi(i)-1) + LDS(p, i))$$

We can compute $LDS(p, j)$ for all $j > p$ in $O(n \log \log n)$ time. Hence this algorithm runs in $O(n^2 \log \log n)$ time.

Chapter 7

Pattern Avoidance: Combinatorial Results

Here we are interested in *Extremal Bounds.*

Let A_1, A_2 be two permutation classes. Then define $ext(A_1, A_2)$ as follows:

$$ext(A_1, A_2) = \min_{\pi \in A_2} \max_{\substack{\rho \in A_1 \\ \rho \leq \pi}} |\rho|$$

Also in the following table let M be the set of all monotone permutations. Let G be any permutation class which admits permutations whose graphs are $\sqrt{n}$-partite.

A_1	A_2	Bounds
X st $M \subseteq X$	Y	$\Omega(\sqrt{n})$
Separable	Baxter	$\Omega(\sqrt{n})$
Stack Sortable	Separable	$\Theta(\sqrt{n})$
Stack Sortable	G	$O(\sqrt{n})$
Jordan	(Arbitrary)	$\sim \frac{5n}{6}$

Let S = Separable, B = Baxter.

Conjecture 7.1. $ext(S, B) \in \Omega(n)$

7.1 Stack Sortable in Separable

Theorem 7.2. *For separable permutations π, the size of the largest $\rho \in Av(231)$ we can guarantee such that $\rho \leq \pi$ is in $\Theta(\sqrt{n})$.*

Proof. To prove that this number is $\Omega(\sqrt{n})$ recall that any permutation contains an increasing/decreasing permutation of size $\geq \sqrt{n}$. A monotone permutation is stack sortable.

Now we exhibit a separable permutation which does not have a stack sortable permutation of size $\omega(\sqrt{n})$. The class of permutation graphs of separable permutations are Cographs (P_4-free) while the class for Stack Sortable are (P_4, C_4)-free graphs. Consider G, a balanced $\sqrt{n}$-partite graph.

- G is P_4-free.
- G des not contain a (P_4, C_4)-free graph of size $\geq 2\sqrt{n}$.

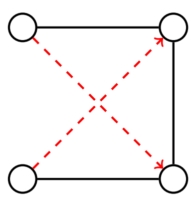

Figure 7.1: Missing edges in P_4

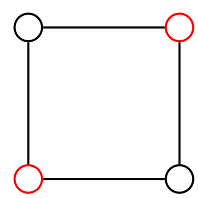

Figure 7.2: A $K_{2,2}$ graph is a C_4

Here we will be assuming a labelling of the classes in the partition of the graph from the set $[\sqrt{n}]$.

- By way of contradiction suppose there was a P_4 in the graph. The vertices at the ends must belong to the same class, say k_1. Now the remaining vertices must belong to a different class than k_1, say k_2, k_3 such taht $k_1 \neq k_2$ and $k_1 \neq k_3$. But then there are edges missing form this graph, as illustrated in Fig 7.1. $\Rightarrow\Leftarrow$
- If we pick 4 vertices such that 2 of them belong to class k_1 and 2 of them belong to class k_2, such that $k_1 \neq k_2$ then we induce a C_4, as illustrated in Fig 7.2. Therefore let us say we pick at least 2 vertices from a partition, since we want to maximise let us pick the entire partition. From the remaining partition, we can only pick atmost 1 vertex. Hence we have $\sqrt{n} + (\sqrt{n} - 1) = 2\sqrt{n} - 1$ vertices. We cannot do better.

□

7.2 Jordan

Using the definitions of Kozma [8], Jordan permutations are exactly those whose *incidence graphs* are planar. Since all incidence graphs are exactly those which are the union of two Hamiltonian paths, this gives us a lot of structure to work with.

Lemma 7.3. *For any incidence graph, there is no K_5 induced in it.*

Proof. By way of contradiction let us say one existed. Let us see the graph induced by these 5 vertices. This is itself the union of two Hamiltonian paths. Hence, it has atmost $4 + 4 = 8$ edges. But a K_5 must have $\binom{5}{2} = 10$ edges. $\Rightarrow\Leftarrow$ □

Hence we only have to worry about "removing" all $K_{3,3}$ from our graph to get a planar graph.

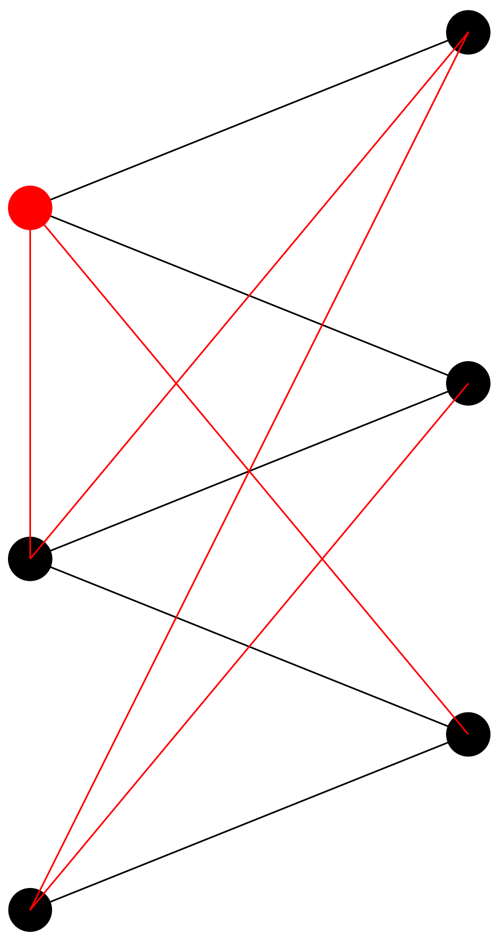

Figure 7.3: $K_{3,3}$ plus an edge can be thought of as union of two Hamiltonian paths

Theorem 7.4. *For any permutation of size n, there is a Jordan permutation of size atmost $\frac{5n}{6}$ induced in it.*

Proof. Let us see how we can get a $K_{3,3}$ in our graph. Since the graph induced on these 6 vertices will have exactly 10 edges and a $K_{3,3}$ has 9 edges, we always have an extra edge. Let us choose one of the paths, divide the vertex set into a bipartition on the basis of these edges. Now, let us remove the one edge from the other path which distrubs the bipartiteness of this graph, call it e_0. We are left a particular induced graph as illustrated in Fig 7.3.

Now you have 4 degree 1 vertices, and e_0 could have connected a pair of them such that we get a path (we have 4 choices). But if we delete one of the endpoints of e_0 from the graph, these 6 vertices can't participate in any other $K_{3,3}$. Hence we can inductively show a lower bound of $\frac{5n}{6}$. □

Notice that this also gives us a characterisation of Jordan permutations, although not a very neat one. We can avoid all the patterns which give rise to a $K_{3,3}$.

Lemma 7.5. *Jordan permutations are exactly*
$Av(416352, 253614, 361452, 254163, 361425, 524163, 416352, 253614)$

7.3 Number of Wedge Permutations

Let $w(n)$ denote the number of Wedge permutations ($Av(213, 231)$) of size n. Neou showed in [9] that every Wedge permutation begins with either a 1 or a n. Now neither 1 nor n can be involved in either 213 or 231 if they are at the first index. Hence the remaining permutation when reduced to $[n-1]$ is also a wedge permutation and every wedge permutation is a valid suffix. This gives us a recurrence.

$$w(n) = 2w(n-1); w(1) = 1$$

$$\Rightarrow w(n) = 2^{n-1}$$

Bibliography

[1] Abed, F., Chalermsook, P., Correa, J., Karrenbauer, A., Pérez-Lantero, P., Soto, J. A., and Wiese, A. On Guillotine Cutting Sequences. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)* (Dagstuhl, Germany, 2015), N. Garg, K. Jansen, A. Rao, and J. D. P. Rolim, Eds., vol. 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 1–19.

[2] Ackerman, E., Barequet, G., and Pinter, R. Y. A bijection between permutations and floorplans, and its applications. *Discrete Applied Mathematics 154*, 12 (2006), 1674 – 1684.

[3] Bespamyatnikh, S., and Segal, M. Enumerating longest increasing subsequences and patience sorting, 2000.

[4] Bose, P., Buss, J. F., and Lubiw, A. Pattern matching for permutations. *Information Processing Letters 65*, 5 (1998), 277 – 283.

[5] Bouvel, M., Guerrini, V., Rechnitzer, A., and Rinaldi, S. Semi-Baxter and strong-Baxter: two relatives of the Baxter sequence. *ArXiv e-prints* (Feb. 2017).

[6] Fox, J. Stanley-wilf limits are typically exponential. *CoRR abs/1310.8378* (2013).

[7] Guillemot, S., and Marx, D. Finding small patterns in permutations in linear time. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2014), SODA '14, Society for Industrial and Applied Mathematics, pp. 82–101.

[8] Kozma, L. Faster and simpler algorithms for finding large patterns in permutations. *CoRR abs/1902.08809* (2019).

[9] Neou, B. E. *Permutation pattern matching*. PhD thesis, Document and Text Processing. Universit Paris-Est, 2017.

[10] OEIS Foundation Inc. (2018). *The On-Line Encyclopedia of Integer Sequences*. http://oeis.org/.

[11] Pach, J., and Tardos, G. Cutting glass. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 2000), SCG '00, ACM, pp. 360–369.

[12] Silvanus, J., and Vygen, J. Few Sequence Pairs Suffice: Representing All Rectangle Placements. *ArXiv e-prints* (Aug. 2017).

[13] Young, E. F. Y., Chu, C. C. N., and Shen, C. Twin binary sequences: A non-redundant representation for general non-slicing floorplan. In *Proceedings of the 2002 International Symposium on Physical Design* (New York, NY, USA, 2002), ISPD '02, ACM, pp. 196–201.