Pattern Avoidance for k-Catalan Sequences

Aaron Williams College

Permutation Patterns 2023

Introduction

This is my 1^{st} Permutation Patterns Conference!

I didn't plan to study permutation patterns, but ...

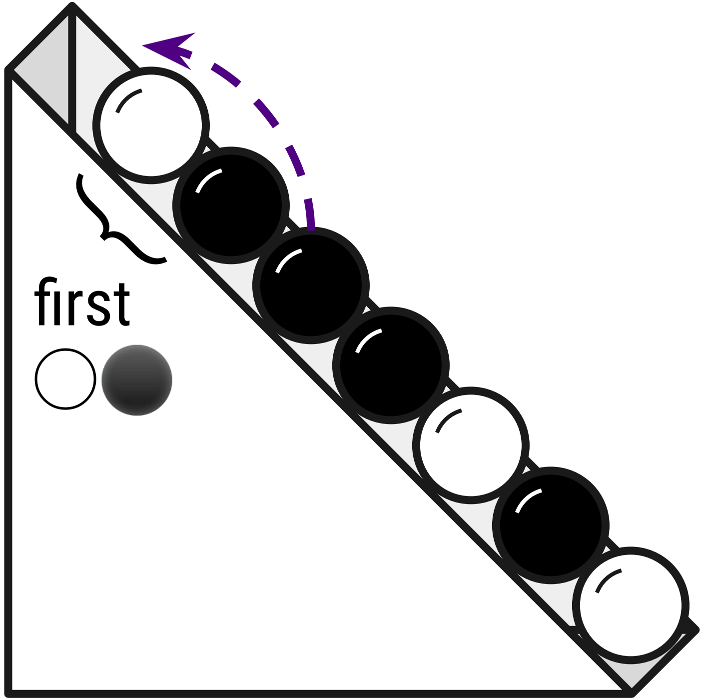
Michael Albert (2013)

UNIVERSITY of OTAGO
Te Whare Wānanga o Otāgo
NEW ZEALAND
SAPERE AUDE

Permutation patterns **suck in** researchers from other areas of mathematics and computer science.

Gray Codes and Combinatorial Generation

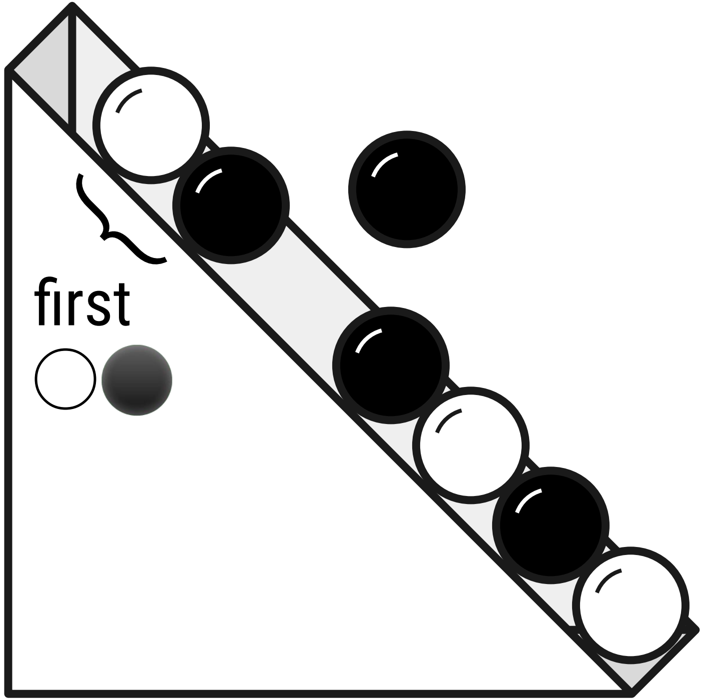
Task: Rearrange the marbles ●●●●○○○ in all $\binom{n}{w} = \binom{7}{3}$ ways.



Solution: Repeatedly pull up the marble after the first ○● pair. If there is no such marble, then pull up the last marble.

Gray Codes and Combinatorial Generation

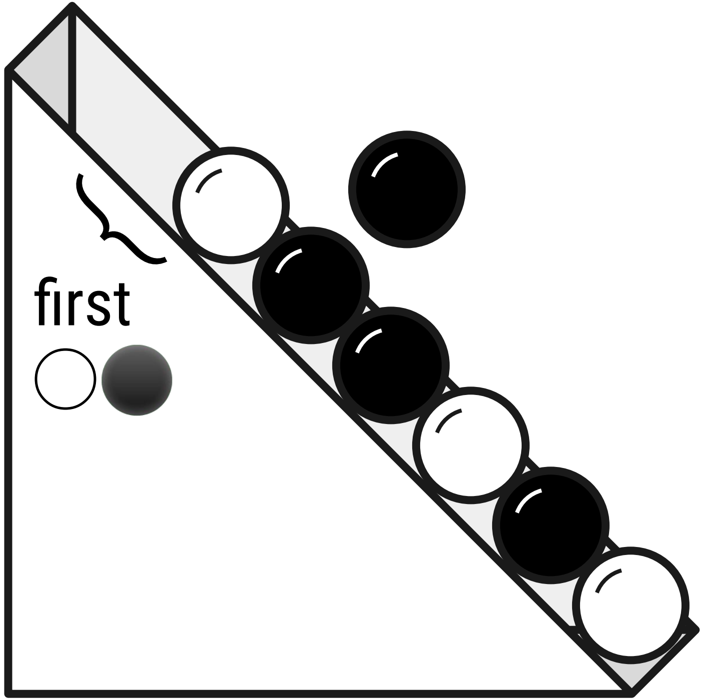
Task: Rearrange the marbles ●●●●○○○ in all $\binom{n}{w} = \binom{7}{3}$ ways.



Solution: Repeatedly pull up the marble after the first ○● pair. If there is no such marble, then pull up the last marble.

Gray Codes and Combinatorial Generation

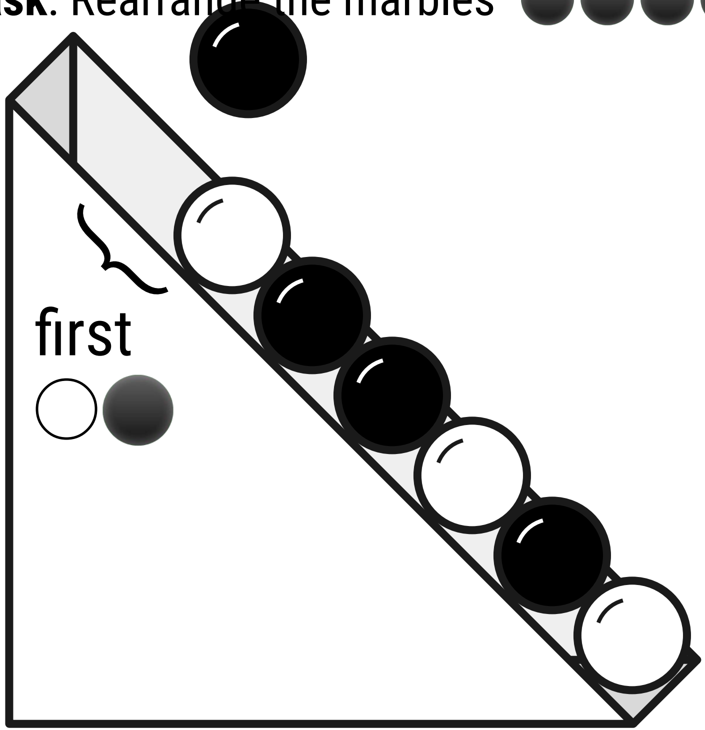
Task: Rearrange the marbles ●●●●○○○ in all $\binom{n}{w} = \binom{7}{3}$ ways.



Solution: Repeatedly pull up the marble after the first ○● pair. If there is no such marble, then pull up the last marble.

Gray Codes and Combinatorial Generation

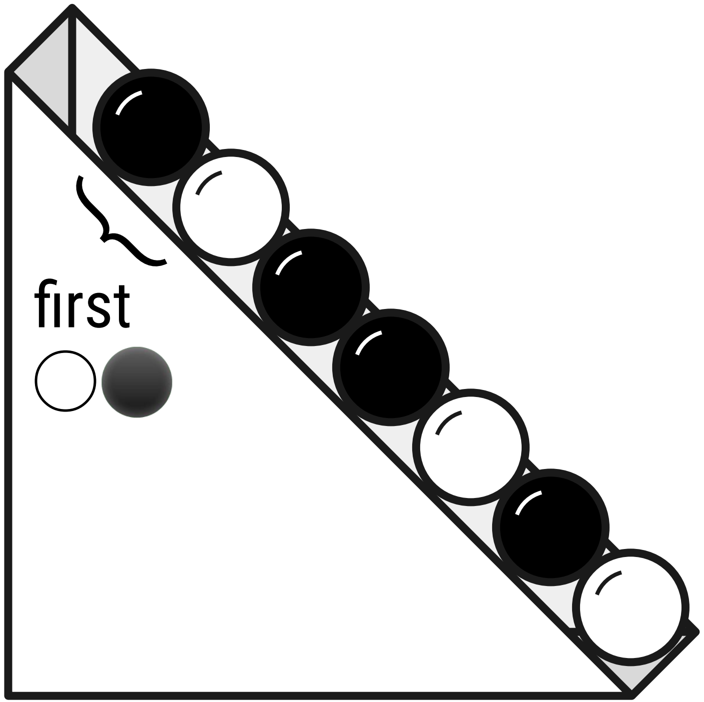
Task: Rearrange the marbles in all $\binom{n}{w} = \binom{7}{3}$ ways.



Solution: Repeatedly pull up the marble after the first pair. If there is no such marble, then pull up the last marble.

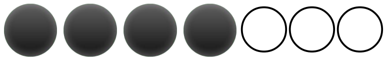
Gray Codes and Combinatorial Generation

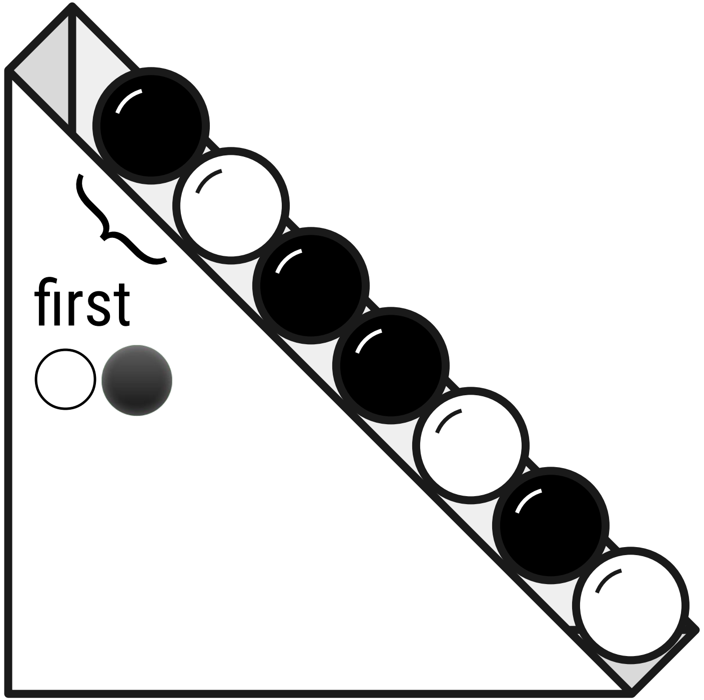
Task: Rearrange the marbles ●●●●○○○ in all $\binom{n}{w} = \binom{7}{3}$ ways.



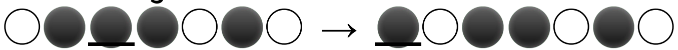
Solution: Repeatedly pull up the marble after the first ○● pair. If there is no such marble, then pull up the last marble.

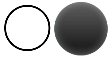
Gray Codes and Combinatorial Generation

Task: Rearrange the marbles  in all $\binom{n}{w} = \binom{7}{3}$ ways.



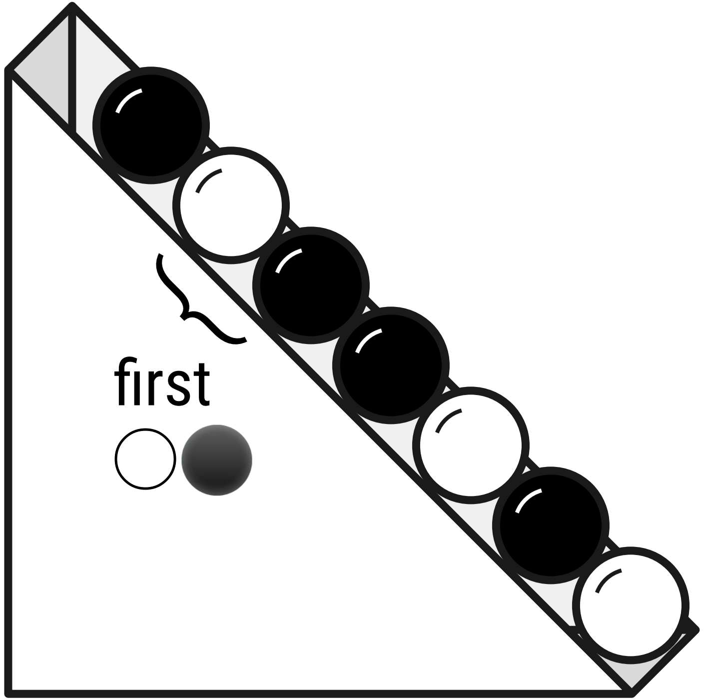
Turning one combination into the next.



Solution: Repeatedly pull up the marble after the first  pair. If there is no such marble, then pull up the last marble.

Gray Codes and Combinatorial Generation

Task: Rearrange the marbles ●●●●○○○ in all $\binom{n}{w} = \binom{7}{3}$ ways.

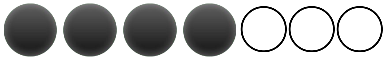


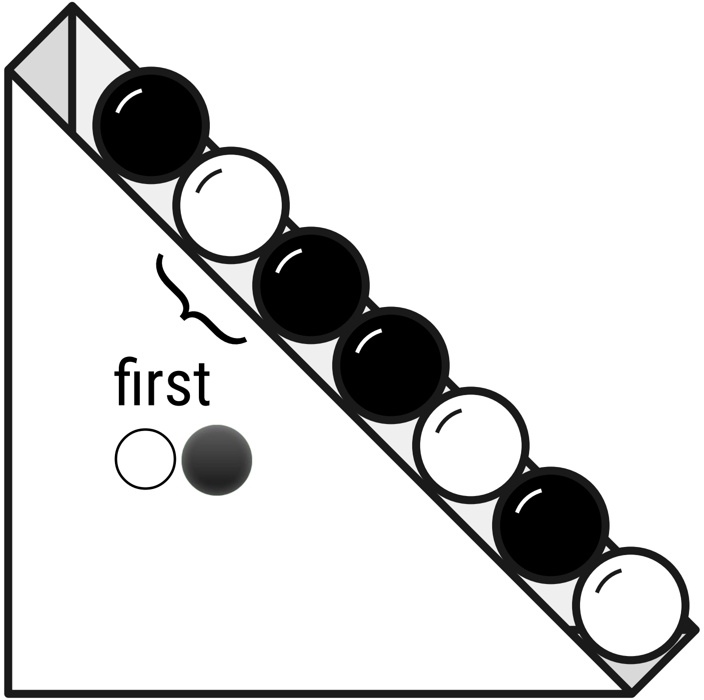
Turning one combination into the next.

○●●●○●○ → ●○●●○●○

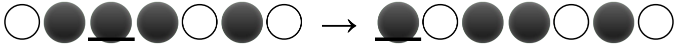
Solution: Repeatedly pull up the marble after the first ○● pair. If there is no such marble, then pull up the last marble.

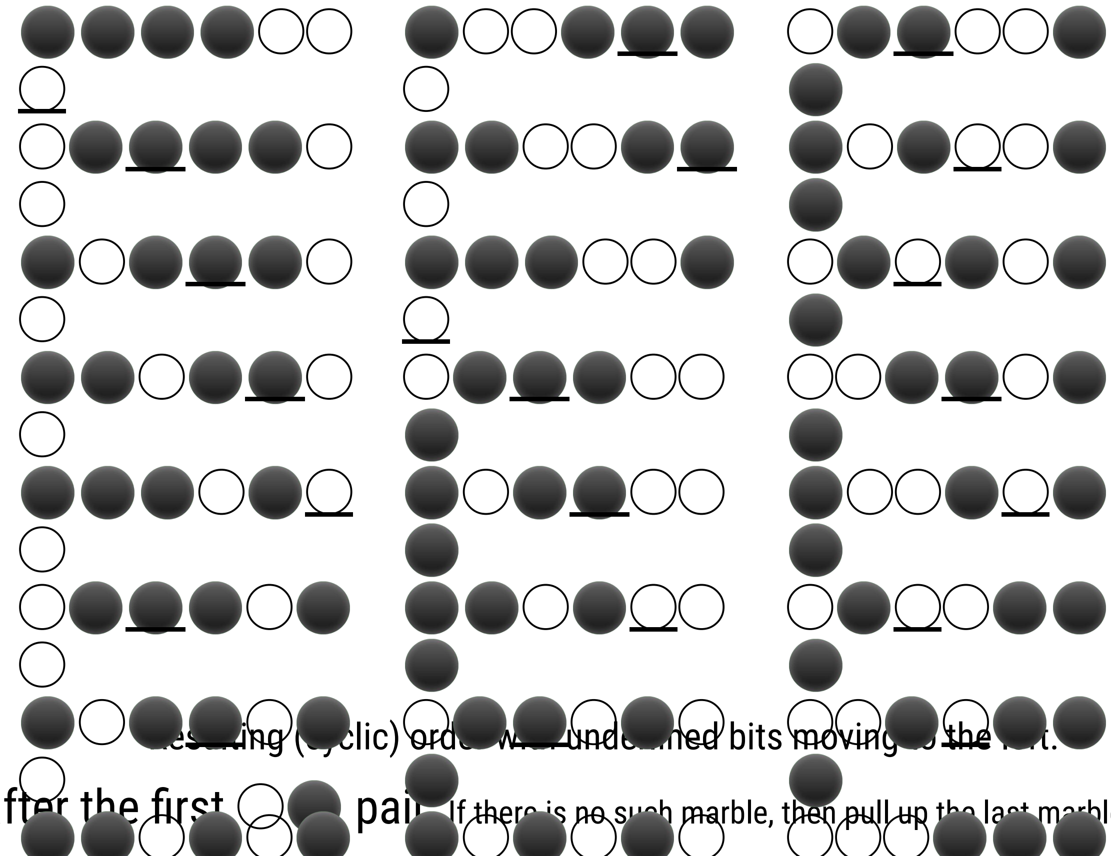
Gray Codes and Combinatorial Generation

Task: Rearrange the marbles  in all $\binom{n}{w} = \binom{7}{3}$ ways.



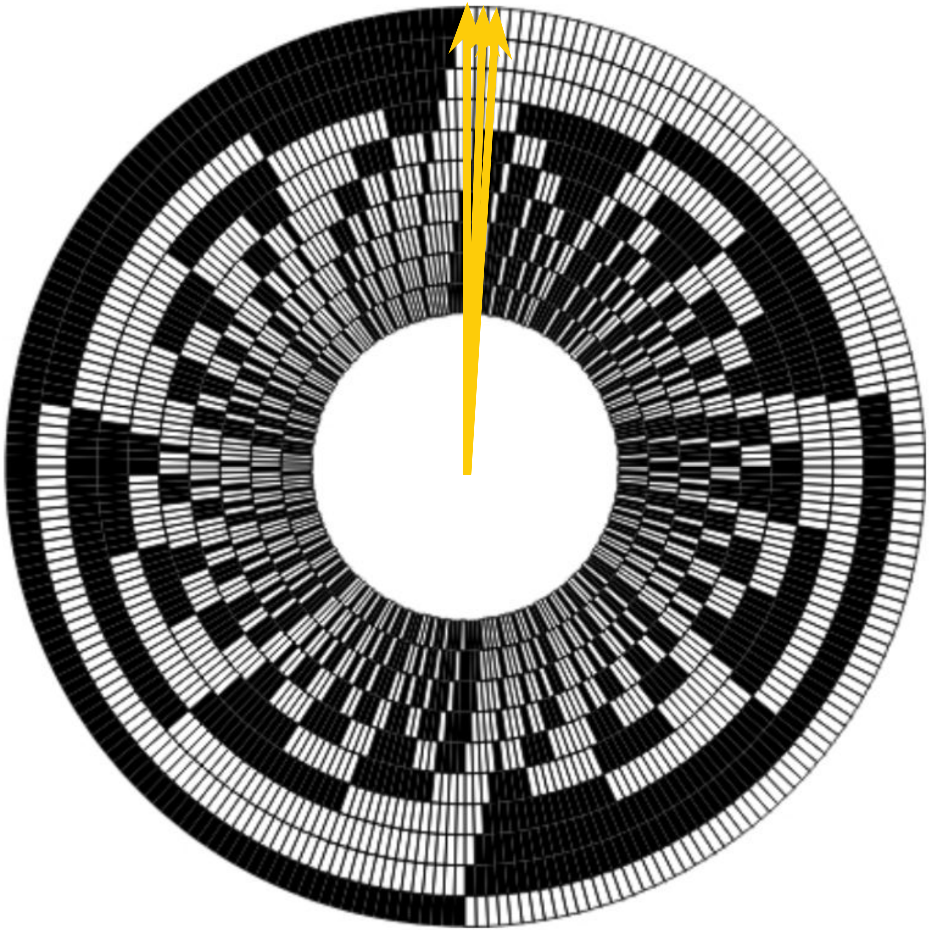
Turning one combination into the next.

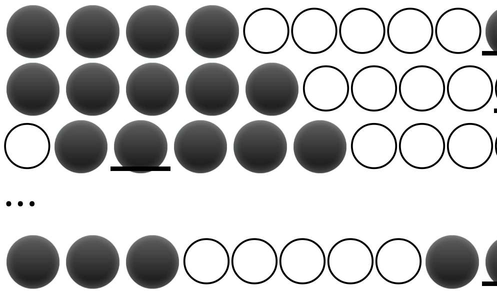




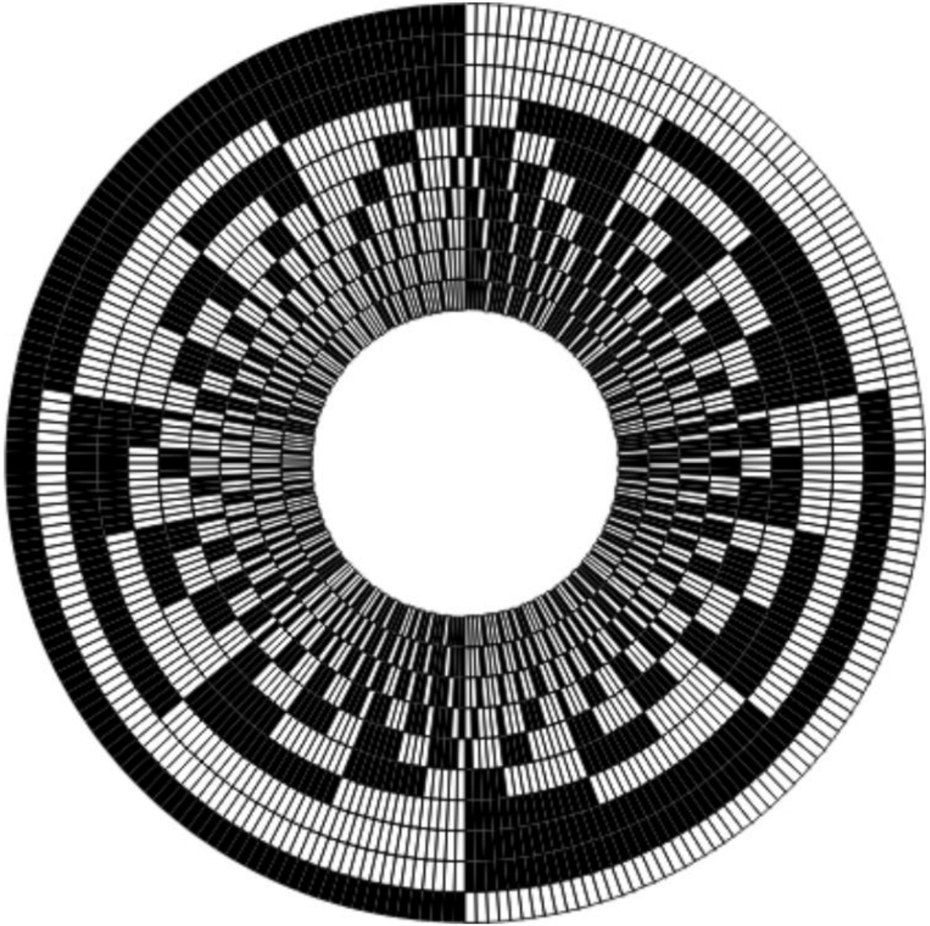
Resulting (cyclic) order with underlined bits moving to the left.

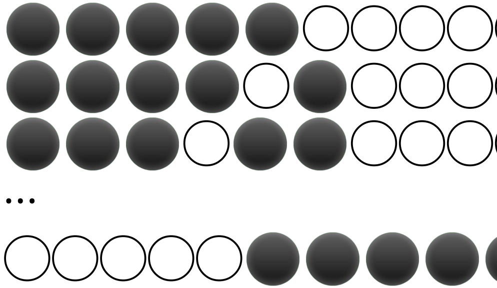
Solution: Repeatedly pull up the marble after the first pair. If there is no such marble, then pull up the last marble.





Cool-lex order for the combinations of $n = 10$ bits with $w = 5$ white.





Co-lexicographic order.

```python
# Input:  Positive integers w, b.
# Output: Every (n,w)-combination is yielded as a tuple where n = w+b.
def coolCombinations(w,b):
    n = w + b
    combo = (1,)*w + (0,)*b
    inc = n-2
    while inc != n-1:
        index = min(inc+2,n-1)
        combo = (combo[index],) + combo[:index] + combo[index+1:]
        inc = 0 if combo[0] < combo[1] else inc+1
        yield combo
```

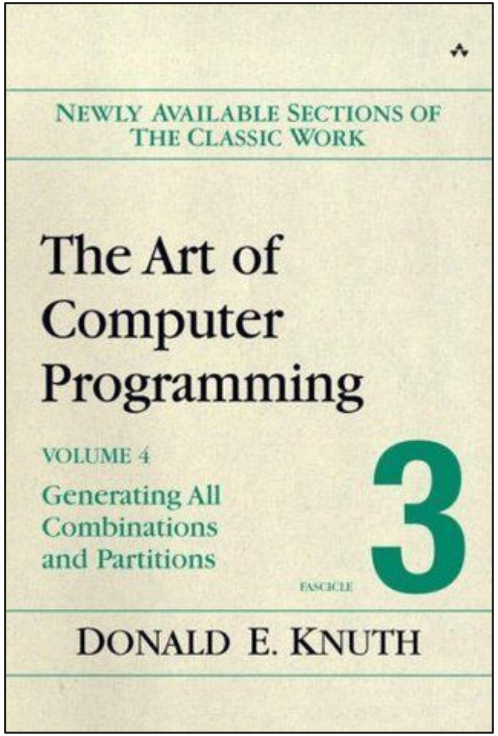
At most two pairs of bits change.

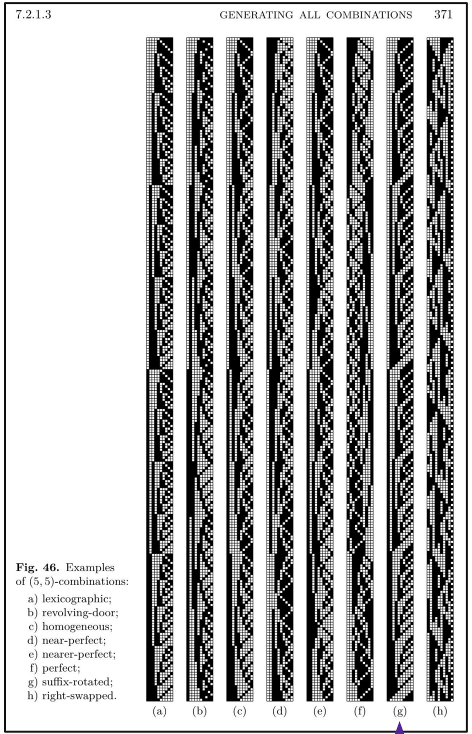
```python
# Input:  A multiset of symbols (with at least two distinct symbols).
# Output: Every permutation of the multiset is yielded as a tuple.
def coolMultiperms(multiset):
    perm = tuple(sorted(multiset, reverse=True))
    n = len(perm)
    inc = n-2
    while inc != n-1:
        index = inc+1 if (inc == len(perm)-2 or perm[inc] < perm[inc+2]) else inc+2
        perm = (perm[index],) + perm[:index] + perm[index+1:]
        inc = 0 if perm[0] < perm[1] else inc+1
        yield perm
```

Generalization to multiset permutations

$\textsc{CoolCombo}(s,t)$
- $n \leftarrow s+t$
- $b \leftarrow 1^t 0^s$
- $x \leftarrow t$
- $y \leftarrow t$
- $\texttt{visit}(b)$
- **while** $x < n$ **do**
  - $b_x = 0$
  - $b_y = 1$
  - $x \leftarrow x+1$
  - $y \leftarrow y+1$
  - **if** $b_x = 0$ **then**
    - $b_x \leftarrow 1$
    - $b_1 \leftarrow 0$
    - **if** $b_2 = 1$ **then**
      - $x \leftarrow 2$
      - $y \leftarrow 1$
  - $\texttt{visit}(b)$

Python code (list-based) and pseudocode (array-based) for generating cool-lex order of combinations and multiset permutations.





Python code (list-based) and pseudocode (array-based) for generating cool-lex order of combinations and multiset permutations.

Discrete Mathematics
Volume 309, Issue 17, 6 September 2009, Pages 5305-5320

multiset permutations

The coolest way to generate ~~combinations~~

Frank Ruskey, Aaron Williams

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.disc.2007.11.048 Get rights and content

Abstract

We present a practical and elegant method for generating all (s,t)-combinations (binary strings with s zeros and t ones): Identify the shortest prefix ending in 010 or 011 (or the entire string if no such prefix exists), and rotate it by one position to the right. This iterative rule gives an order to (s,t)-combinations that is circular and genlex. Moreover, the rotated portion of the string always contains at most four contiguous runs of zeros and ones, so every iteration can be achieved by transposing at most two pairs of bits. This leads to an efficient loopless and branchless implementation that consists only of two variables and six assignment statements. The order also has a number of striking similarities to colex order, especially its recursive definition and ranking algorithm. In the

SODA 2009

ELSEVIER

Discrete Mathematics

Volume 309, Issue 17, 6 September 2009, Pages 5305-5320

multiset permutations

The coolest way to generate ~~combinations~~

Loopless Generation of Multiset Permutations using a Constant Number of Variables by Prefix Shifts

Aaron Williams *

Abstract

This paper answers the following mathematical question: Can multiset permutations be ordered so that each permutation is a prefix shift of the previous permutation? Previously, the answer was known for the permutations of any set, and the permutations of any multiset whose corresponding set contains only two elements. This paper also answers the following algorithmic question: Can multiset permutations be generated by a loopless algorithm that uses sublinear additional storage? Previously, the best loopless algorithm used a linear amount of additional storage. The answers to these questions are both yes.

minimal-change order. A minimal-change order is an order in which each successive object can be obtained from the previous by making one small modification of a certain type. The existence or non-existence of minimal-change orders depend upon the type of object and the type of modification. New results in this area are often quite difficult to find, but the results that are found tend to be elegant and simple. The mathematical question answered in this paper is the following.

QUESTION 1. Can multiset permutations be ordered so that each permutation is a prefix shift of the previous permutation?

similarities to colex order, especially its recursive definition and ranking algorithm. In the

SODA 2009

1+ million downloads (2021–2023)

Useful for Research in Permutation Patterns!

RDocumentation

Search all packages and functions

multicool

COPY LINK

https://rdocumentation.org/packages/multicool/versions/0.1-12

VERSION

0.1-12

INSTALL

```
install.packages('multicool')
```

MONTHLY DOWNLOADS

30,528

VERSION	LICENSE
0.1-12	GPL-2

MAINTAINER	LAST PUBLISHED
James Curran	June 29th, 2021

ultipermute

npm install multipermute

build failing

efficient multiset permutations

Overview

This package contains a method to generate all permutations of a multiset. The method is described in "Loopless Generation of Multiset Permutations using a Constant Number of Variables by Prefix Shifts." Aaron Williams, 2009. To quote Aaron's website:

> The permutations of any multiset are generated by this simple rule:
>
> - The first symbol a is shifted to the right until it passes over consecutive symbols b c with b < c.
> - If a > b, then a is inserted after b; otherwise, if a <= b, then a is inserted after c.
> - (If there is no such b c then a is shifted until it passes over the rightmost symbol.)
>
> This result leads to the first O(1)-time / O(1)-additional variable algorithm for generating the permutations of a multiset (see publication in SODA 2009)."

In other words, this algorithm is so awesome that it hurts.

COS ++

Home

Permutations

Permutations

Multiset permutations

Colored permutations

Meanders

Pattern avoidance

Binary trees

Rectangulations

Elimination trees

Acyclic orientations

Bitstrings

Graphs

Partitions

Necklaces

Generate permutations of a multiset

Generate all permutations of a multiset. Provide the multiset by specifying its frequency sequence, e.g. "1 1 3 2" for the multiset $\{1, 2, 3, 3, 3, 4, 4\}$.

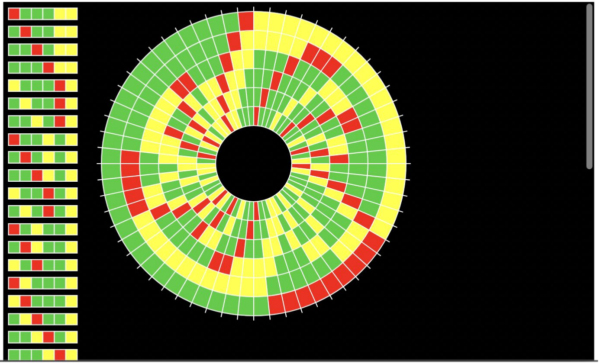
Frequency sequence: 1 2 3 (max. 20)

Algorithm: prefix rotation (Williams' cool-lex)

Output format: graphics (≤500 objects)

Generate

Output numbering graphics



Torsten Mütze's COS++

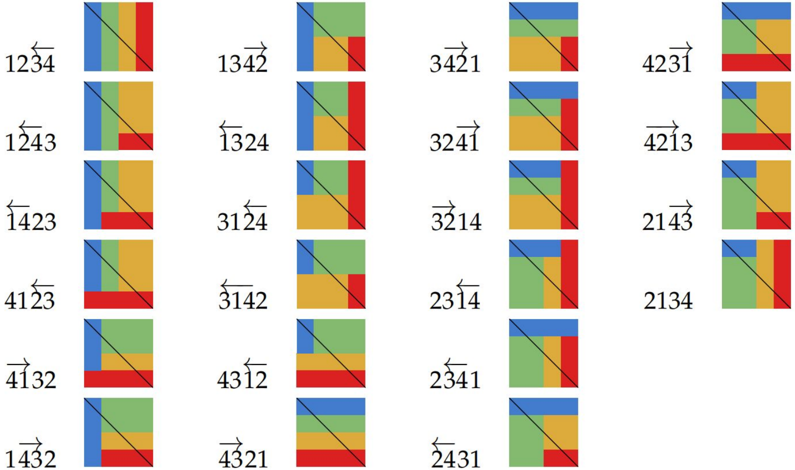
combos.org graycodes.com

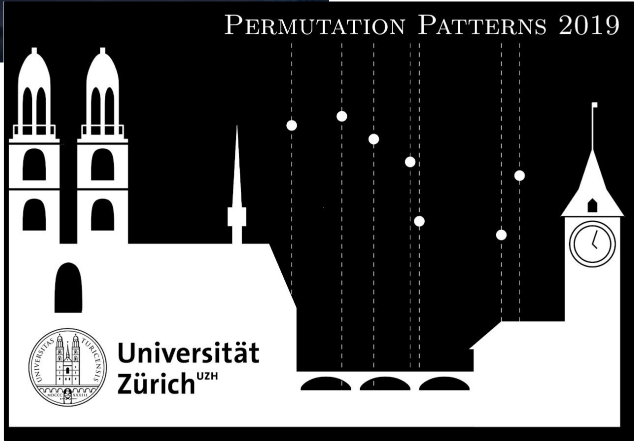
Multiset permutations usually are not generated by standard libraries (e.g., `itertools` in Python).

The cool-lex successor rule has been implemented in various languages by various people.

Let's Gray code
Av(2413, 3142).

Torsten Mütze (2018)



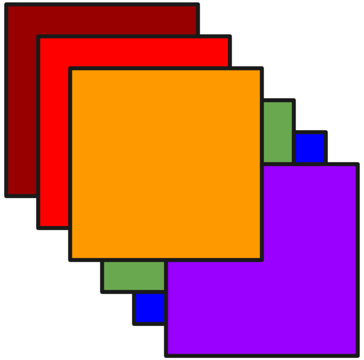


Combinatorial generation via permutation languages (I – VI) series *started* to suck me in!

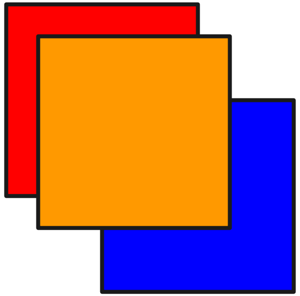
Catalan Squares

A new Catalan object emphasizing layering and obfuscation.

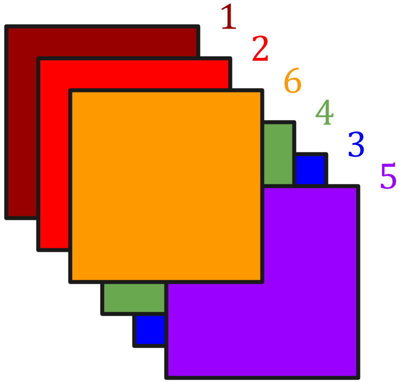
- There are n squares layered along the main diagonal. All of the squares touch a common position (i.e., overlap).
- If a rectangle is in front of rectangles to its left and right, then their relative layering is hidden. By convention, we assume that the left rectangles are behind the right rectangles, and this fixes the stacking order.



Catalan squares instance for $n = 6$. The orange rectangle is the frontmost.



The orange rectangle hides the relative layering of the red and blue rectangles.

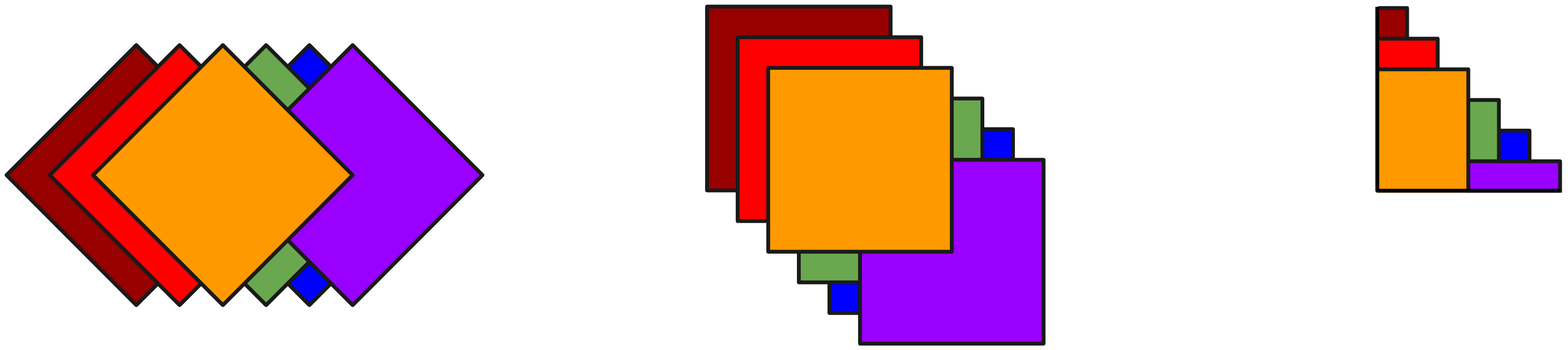


Representation as a permutation.

126435

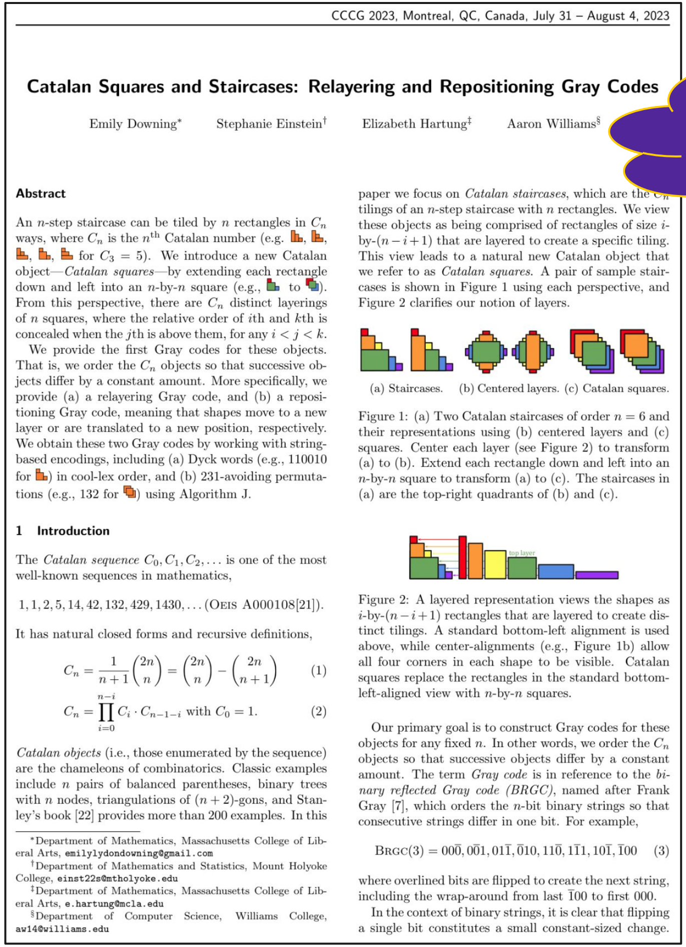
Each instance can be represented by a permutation.

- The permutation avoids 231. Such a pattern would indicate a left rectangle (i.e., 2) in front of a right rectangle (i.e., 1) relative to a middle rectangle (i.e., 3) that is in front of both.



Notes:

- Catalan squares could be represented using diamonds (or other shapes) organized left-to-right.
- The top-right quadrant is a layer-based representation of a Catalan staircase.


CCCG 2023, Montreal, QC, Canada, July 31 – August 4, 2023

Catalan Squares and Staircases: Relayering and Repositioning Gray Codes

Emily Downing* Stephanie Einstein† Elizabeth Hartung‡ Aaron Williams§

Abstract

An n-step staircase can be tiled by n rectangles in C_n ways, where C_n is the n^{th} Catalan number (e.g. for $C_3 = 5$). We introduce a new Catalan object—*Catalan squares*—by extending each rectangle down and left into an n-by-n square. From this perspective, there are C_n distinct layerings of n squares, where the relative order of ith and kth is concealed when the jth is above them, for any $i < j < k$.

We provide the first Gray codes for these objects. That is, we order the C_n objects so that successive objects differ by a constant amount. More specifically, we provide (a) a relayering Gray code, and (b) a repositioning Gray code, meaning that shapes move to a new layer or are translated to a new position, respectively. We obtain these two Gray codes by working with string-based encodings, including (a) Dyck words (e.g., 110010 for) in cool-lex order, and (b) 231-avoiding permutations (e.g., 132 for) using Algorithm J.

1 Introduction

The *Catalan sequence* $C_0, C_1, C_2, \ldots$ is one of the most well-known sequences in mathematics,

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, \ldots \text{ (OEIS A000108[21])}.$$

It has natural closed forms and recursive definitions,

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n+1} \quad (1)$$

$$C_n = \prod_{i=0}^{n-i} C_i \cdot C_{n-1-i} \text{ with } C_0 = 1. \quad (2)$$

Catalan objects (i.e., those enumerated by the sequence) are the chameleons of combinatorics. Classic examples include n pairs of balanced parentheses, binary trees with n nodes, triangulations of $(n+2)$-gons, and Stanley's book [22] provides more than 200 examples. In this paper we focus on *Catalan staircases*, which are the C_n tilings of an n-step staircase with n rectangles. We view these objects as being comprised of rectangles of size i-by-$(n-i+1)$ that are layered to create a specific tiling. This view leads to a natural new Catalan object that we refer to as *Catalan squares*. A pair of sample staircases is shown in Figure 1 using each perspective, and Figure 2 clarifies our notion of layers.

(a) Staircases. (b) Centered layers. (c) Catalan squares.

Figure 1: (a) Two Catalan staircases of order $n = 6$ and their representations using (b) centered layers and (c) squares. Center each layer (see Figure 2) to transform (a) to (b). Extend each rectangle down and left to an n-by-n square to transform (a) to (c). The staircases in (a) are the top-right quadrants of (b) and (c).

top layer

Figure 2: A layered representation views the shapes as i-by-$(n-i+1)$ rectangles that are layered to create distinct tilings. A standard bottom-left alignment is used above, while center-alignments (e.g., Figure 1b) allow all four corners in each shape to be visible. Catalan squares replace the rectangles in the standard bottom-left-aligned view with n-by-n squares.

Our primary goal is to construct Gray codes for these objects for any fixed n. In other words, we order the C_n objects so that successive objects differ by a constant amount. The term *Gray code* is in reference to the *binary reflected Gray code (BRGC)*, named after Frank Gray [7], which orders the n-bit binary strings so that consecutive strings differ in one bit. For example,

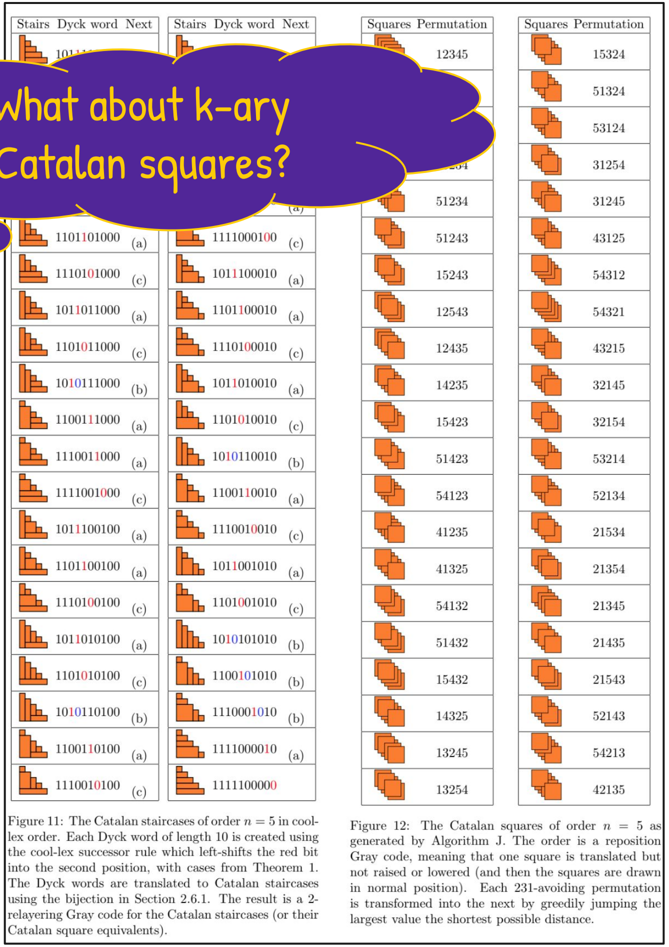
$$\text{BRGC}(3) = 00\bar{0}, 0\bar{0}1, 01\bar{1}, \bar{0}10, 11\bar{0}, 1\bar{1}1, 10\bar{1}, \bar{1}00 \quad (3)$$

where overlined bits are flipped to create the next string, including the wrap-around from last $\bar{1}00$ to first 000.

In the context of binary strings, it is clear that flipping a single bit constitutes a small constant-sized change.

*Department of Mathematics, Massachusetts College of Liberal Arts, emilylydondowning@gmail.com
†Department of Mathematics and Statistics, Mount Holyoke College, einst22s@mtholyoke.edu
‡Department of Mathematics, Massachusetts College of Liberal Arts, e.hartung@mcla.edu
§Department of Computer Science, Williams College, aw14@williams.edu

What about k-ary Catalan squares?



Dyck word	Next	Dyck word	Next
1011[illegible]		[illegible]	
[illegible]		[illegible]	
[illegible]		[illegible]	
[illegible]		[illegible]	
[illegible]		[illegible]	
[illegible]		[illegible]	(a)
1101101000	(a)	1111000100	(c)
1110101000	(c)	1011100010	(a)
1011011000	(a)	1101100010	(a)
1101011000	(c)	1110100010	(c)
1010111000	(b)	1011010010	(a)
1100111000	(a)	1101010010	(c)
1110011000	(a)	1010110010	(b)
1111001000	(c)	1100110010	(a)
1011100100	(a)	1110010010	(c)
1101100100	(a)	1011001010	(a)
1110100100	(c)	1101001010	(c)
1011010100	(a)	1010101010	(b)
1101010100	(c)	1100101010	(b)
1010110100	(b)	1110001010	(b)
1100110100	(a)	1111000010	(a)
1110010100	(c)	1111100000	

Figure 11: The Catalan staircases of order $n = 5$ in cool-lex order. Each Dyck word of length 10 is created using the cool-lex successor rule which left-shifts the red bit into the second position, with cases from Theorem 1. The Dyck words are translated to Catalan staircases using the bijection in Section 2.6.1. The result is a 2-relayering Gray code for the Catalan staircases (or their Catalan square equivalents).

Permutation	Permutation
12345	15324
[illegible]	51324
[illegible]	53124
[illegible]234	31254
51234	31245
51243	43125
15243	54312
12543	54321
12435	43215
14235	32145
15423	32154
51423	53214
54123	52134
41235	21534
41325	21354
54132	21345
51432	21435
15432	21543
14325	52143
13245	54213
13254	42135

Figure 12: The Catalan squares of order $n = 5$ as generated by Algorithm J. The order is a reposition Gray code, meaning that one square is translated but not raised or lowered (and then the squares are drawn in normal position). Each 231-avoiding permutation is transformed into the next by greedily jumping the largest value the shortest possible distance.

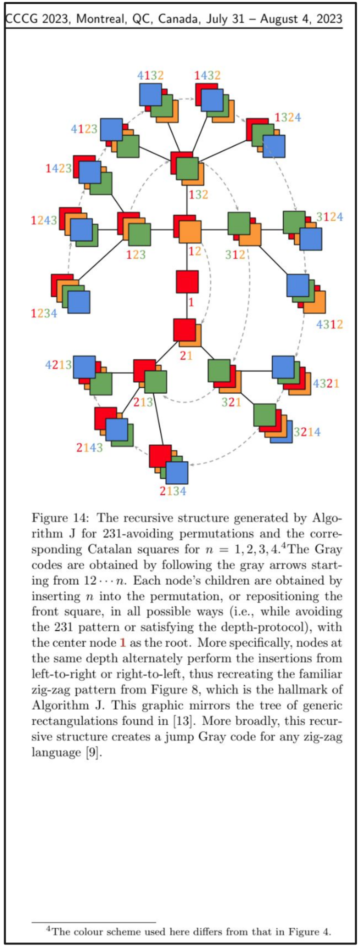

CCCG 2023, Montreal, QC, Canada, July 31 – August 4, 2023

Figure 14: The recursive structure generated by Algorithm J for 231-avoiding permutations and the corresponding Catalan squares for $n = 1, 2, 3, 4$.[4] The Gray codes are obtained by following the gray arrows starting from $12 \cdots n$. Each node's children are obtained by inserting n into the permutation, or repositioning the front square, in all possible ways (i.e., while avoiding the 231 pattern or satisfying the depth-protocol), with the center node 1 as the root. More specifically, nodes at the same depth alternately perform the insertions from left-to-right or right-to-left, thus recreating the familiar zig-zag pattern from Figure 8, which is the hallmark of Algorithm J. This graphic mirrors the tree of generic rectangulations found in [13]. More broadly, this recursive structure creates a jump Gray code for any zig-zag language [9].

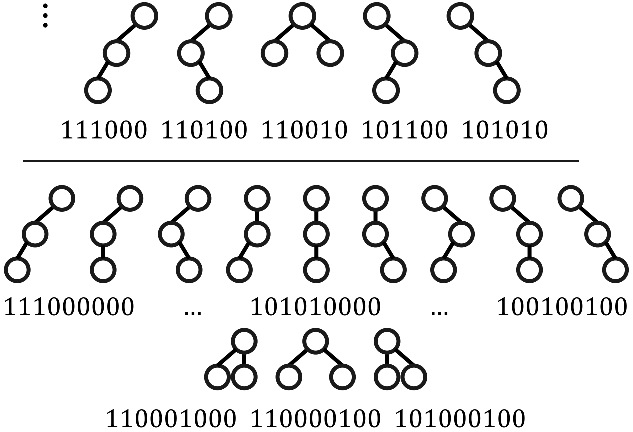
[4] The colour scheme used here differs from that in Figure 4.

Gray codes via cool-lex order for Dyck words and Algorithm J for 231-avoiding permutations (CCCG 2023).

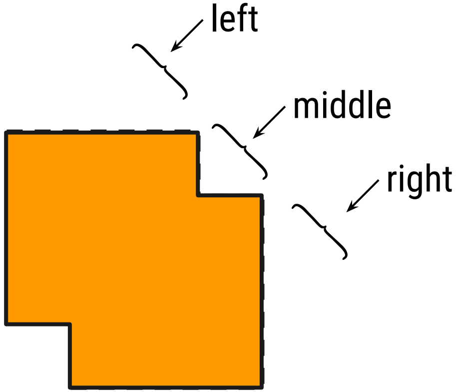
Generalizing Catalan Objects including k-ary Catalan Squares

Many standard (2-ary) Catalan objects have *k-ary* generalizations.

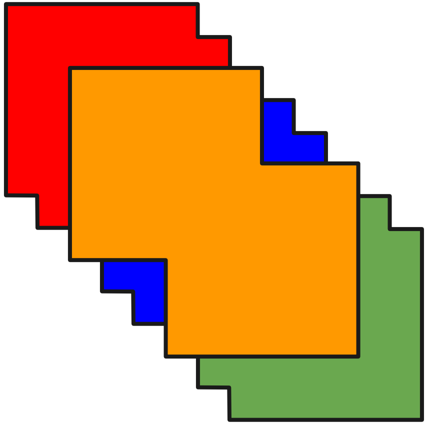
OEIS A000108 (2-Catalan): 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, ...
OEIS A001764 (3-Catalan): 1, 1, 3, 12, 55, 273, 1428, 7752, 43263, 246675, 1430715, 8414640, ...
OEIS A002293 (4-Catalan): 1, 1, 4, 22, 140, 969, 7084, 53820, 420732, 3362260, 27343888,
⋮



Binary trees and Dyck with $n = 3$.
k-ary trees and k-ary Dyck words $k = 3$.



Two squares on a layer instead of one.
More generally, k-1 at the same layer,
which creates k locations for recursion.



Representation using generalized protocol

11422433

Represent a k-ary Catalan square instance using pattern avoidance.

- k-1 copies of each symbol instead of 1 copy.
- Avoids 231 due to our protocol, and ...
- Avoids 121 due to recursive nesting.

THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES ®

A001764

Number of intervals (i.e., ordered pairs (x,y) such that x<=y) in the Kreweras lattice (noncrossing partitions ordered by refinement) of size n, see the Bernardi & Bonichon (2009) and Kreweras (1972) references. - Noam Zeilberger, Jun 01 2016

Number of sum-indecomposable (4231,42513)-avoiding permutations. Conjecturally, number of sum-indecomposable (2431,45231)-avoiding permutations. - Alexander Burstein, Oct 19 2017

a(n) is the number of topologically distinct endstates for the game Planted Brussels Sprouts on n vertices, see Ji and Propp link. - Caleb Ji, May 14 2018

Number of complete quadrillages of 2n+2-gons. See Baryshnikov p. 12. See also Nov. 10 2014 comments in A134264. - Tom Copeland, Jun 04 2018

a(n) is the number of 2-regular words on the alphabet [n] that avoid the patterns 231 and 221. Equivalently, this is the number of 2-regular tortoise-sortable words on the alphabet [n] (see the Defant and Kravitz link). - Colin Defant, Sep 26 2018

a(n) is the number of Motzkin paths of length 3n with n steps of each type, with the condition that (1, 0) and (1, 1) steps alternate (starting with (1, 0)). - Helmut Prodinger, Apr 08 2019

a(n) is the number of uniquely sorted permutations of length 2n+1 that avoid the patterns 312 and 1342. - Colin Defant, Jun 08 2019

The compositional inverse o.g.f. pair in Copeland's comment above are

OEIS A001764 does not mention $\mathrm{Av}^2(231, 121)$ (i.e., 2-regular words avoiding 231 and 121).
But it does discuss $\mathrm{Av}^2(231, 122)$ … **Now I'm sucked in!**

Colin Defant <colindefant@gmail.com> Wed, Jan 18, 7:11 PM

to me

Hi Aaron,

I agree that this is interesting and deserves to be in print somewhere, and I think it would be a great submission to Permutation Patterns! I don't know of anywhere in the literature where it's written down, so I assume it's fair game. Unfortunately, I'm pretty busy with several projects at the moment, so I don't think I can handle picking up another one. But I wish you the best and hope you get some nice results!

Best,
Colin.

STACK-SORTING FOR WORDS

COLIN DEFANT AND NOAH KRAVITZ

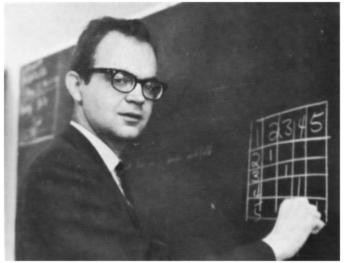
ABSTRACT. We introduce operators **hare** and **tortoise**, which act on words as natural generalizations of West's stack-sorting map. We show that the heuristically slower algorithm **tortoise** can sort words arbitrarily faster than its counterpart **hare**. We then generalize the combinatorial objects known as valid hook configurations in order to find a method for computing the number of preimages of any word under these two operators. We relate the question of determining which words are sortable by **hare** and **tortoise** to more classical problems in pattern avoidance, and we derive a recurrence for the number of words with a fixed number of copies of each letter (permutations of a multiset) that are sortable by each map. In particular, we use generating trees to prove that the ℓ-uniform words on the alphabet $[n]$ that avoid the patterns 231 and 221 are counted by the $(\ell+1)$-Catalan number $\frac{1}{\ell n+1}\binom{(\ell+1)n}{n}$. We conclude with several open problems and conjectures.

AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **77(1)** (2020), Pages 51–68

Generalizing the Catalan Theorem to k-Catalan Theorem

Theorem: The Catalan numbers count the permutations of [n] avoiding 123 (or 231).

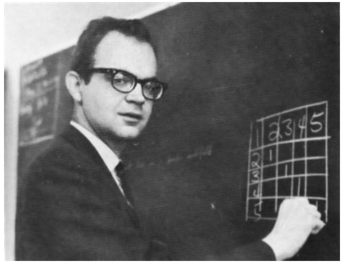
Corollary: $|Av_n(\alpha)| = C_n$ for any choice of $\alpha \in \{123, 132, 213, 231, 312, 321\}$ since (a) $123 \equiv 321$; (b) $132 \equiv 213 \equiv \mathbf{231} \equiv 312$ by reverse and/or inverse.



MacMahon (1915) & Knuth (1968)

Theorem: The Catalan numbers count the permutations of [n] avoiding **123** (or **231**).

Corollary: $|Av_n(\alpha)| = C_n$ for any choice of $\alpha \in \{$**123**, 132, 213, **231**, 312, 321$\}$ since (a) **123** ≡ 321; (b) 132 ≡ 213 ≡ **231** ≡ 312 by reverse and/or inverse.



MacMahon (1915) & Knuth (1968)

Theorem: The k-Catalan numbers count the (k-1)-regular words over [m] avoiding 123 & 112 (or 231 & 121) (or 231 & 221).

Corollary: $|Av^k{}_n(\pi)| = C^k{}_n$ for any *consistent* $\alpha \in \{123, 132, 213, 231, 312, 321\}$ and $\beta \in \{112, 121, 122, 211, 212, 221\}$ by reverse and/or inverse α and β.

Theorem: The k-Catalan numbers count the (k-1)-regular words over [m] avoiding **123** & **112** (or **231** & **121**) (or **231** & **221**).

k = 2

Corollary: $|Av^k_n(\pi)| = C^k_n$ for any *consistent* $\alpha \in \{$**123**, 132, 213, **231**, 312, 321$\}$ and $\beta \in \{$**112**, **121**, 122, 211, 212, **221**$\}$ by reverse and/or inverse α and β.

Reduce α to β by
- $3 \rightarrow 2$ [always]
- $2 \rightarrow 1$ [optional]

Must apply the same symmetries to both patterns.
e.g., Av(123, **132**) ≠ Av(123, **231**)

Defant & Kravitz (2020)
A.W. (2023)

Pairs of Consistent Patterns

There are $6 \cdot 6 = 36$ pairs of patterns of the form (α, β) where

$$\alpha \in \{123, 132, 213, 231, 312, 321\} \text{ and } \beta \in \{112, 121, 122, 211, 212, 221\}.$$

Up to standard symmetries, there are $36 \,/\, 4 = 9$ distinct pairs or *classes*.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
representative	(123,112)	(123,121)	(123,211)	(132,112)	(132,121)	(132,122)	(132,211)	(132,212)	(132,221)
reverse	(321,211)	(321,121)	(321,112)	(231,211)	(231,121)	(231,221)	(231,112)	(231,212)	(231,122)
inverse	(321,221)	(321,212)	(321,122)	(312,221)	(312,212)	(213,112)	(312,122)	(312,121)	(312,112)
both	(123,122)	(123,212)	(123,221)	(213,122)	(213,212)	(312,211)	(213,221)	(213,121)	(213,211)

The 36 pairs of (α, β) patterns partition into 9 classes by reverse and/or inverse.

Our main result proves that 3 of these classes count the k-ary Catalan numbers.

- These classes contain exactly the pairs that are consistent.
- The other classes do not count k-Catalan numbers when $k > 2$.
- The result for C_6 was proven by Defant and Kravitz using $(231, 221)$.

Consistent Pairs and Inequality Chains

The consistent (α, β) pairs can also be understood as enforcing a chain of two inequalities.

- The α pattern enforces two strict inequalities.
- The β pattern softens one of the strict inequalities into a weak inequality.

Avoiding $\alpha = 123$	Avoiding $\beta = 112$	Avoiding $(123, 112)$
$\nexists$ indices $i < j < k$ with values $p_i < p_j < p_k$	$\nexists$ indices $i < j < k$ with values $p_i = p_j < p_k$	$\nexists$ indices $i < j < k$ with values $p_i \leq p_j < p_k$
Avoiding an α pattern in $p_1\, p_2 \cdots p_n$.	Avoiding a β pattern in $p_1\, p_2 \cdots p_n$.	Avoiding a consistent (α, β) pair in $p_1\, p_2 \cdots p_n$.

Similarly, avoiding $(123, 122)$ forbids $p_i < p_j \leq p_k$ instead of $p_i \leq p_j < p_k$.
Of course, weak inequalities only make sense when the word has repeated symbols.

Proofs

Only proving the two new cases …

General Approach

We provide bijections between r-regular words over $[m]$ and $(r+1)$-Dyck words of order m.

- Each word in the first set, $[m^r]$, has m symbols and length $n = rm$.
- Each word in the second set, $\mathbb{D}^{r+1}(m)$, is $(r+1) \cdot m = rm + m = n + m$ bits in length.

Therefore, our bijections cannot be one symbol to one bit.

The mappings must create an additional m bits overall, or $+1$ per element in $[m]$.

111222333444555

A r-regular word over $[m]$ for $r = 3$ and $m = 5$.
It has length $n = r{\cdot}m = 15$.

10001000100010001000

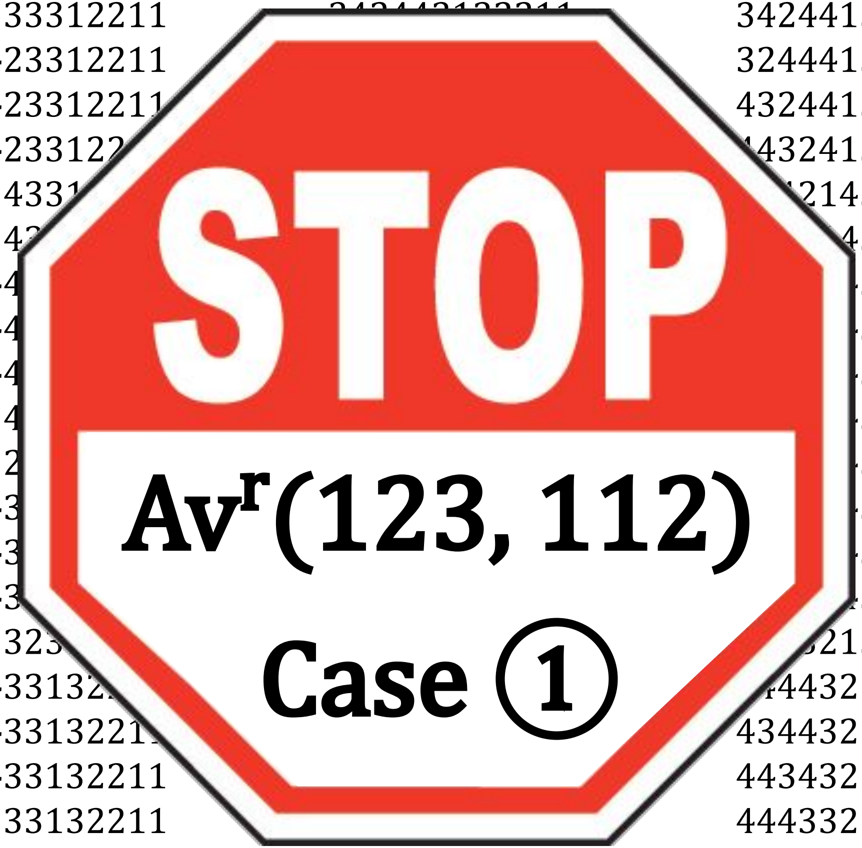
An $(r+1)$-ary Dyck word of order m for $r = 3$ and $m = 5$.
It has length $(r+1){\cdot}m = rm{+}m = n{+}m = 15 + 5 = 20$.

The mappings $f : [m^r] \to \mathbb{D}^{r+1}(m)$ focus on some type of first occurrence for the symbols in $[m]$.

- The other copies of each symbol are used for positioning.
- Each of these first occurrences contribute an additional bit (on average).

144433322211
414433322211
441433322211
444133322211
344413322211
434413322211
443413322211
344143322211
434143322211
341443322211
314443322211
431443322211
443143322211
444313322211
344431322211
434431322211
443431322211
444331322211
344433122211
434433122211
443433122211
444333122211
244433312211
424433312211

442433312211
444233312211
344423312211
43442331221
4434233122
34424331
43424
34244
32444
43244
44324
44432
34443
43443
44343
4443323
244433132
42443313221
442433132211
444233132211
344423132211
434423132211
443423132211
344243132211



434243132211
344421332211
434421332211
443421332211
344241332211

434241332211
342441332211
324441332211
432441332211
443241332211
214332211
4332211
4332211
4332211
4332211
4332211
4332211
4332211
4332211
4332211
21332211
4432132211
434432132211
443432132211
444332132211
344433212211
434433212211
443433212211
444333212211

244433321211
424433321211
442433321211
444233321211
344423321211
434423321211
443423321211
344243321211
434243321211
342443321211
324443321211
432443321211
443243321211
444323321211
344432321211
434432321211
443432321211
444332321211
344433221211
434433221211
443433221211
444333221211

Smallest Seen (aka Left-to-Right Minima)

Given a word $p_1 \, p_2 \cdots p_n$, an entry p_i is *smallest seen* if $p_i = \min(p_1 \, p_2 \cdots p_i) < \min(p_1 \, p_2 \ldots p_{i-1})$. So if the word is read from left-to-right, then it is the first occurrence of a new smallest symbol.

- The first symbol p_1 is always smallest seen.
- An r-regular word in $[m^r]$ has between 1 and m smallest seen entries.

$\underline{3}333\underline{2}222\underline{1}111$

$\underline{3}\underline{2}2233\underline{1}1121$

Words in $[3^4]$ with three smallest seen entries.

$\underline{1}23412341234$

$\underline{1}44433322211$

Words in $[4^3]$ with one smallest seen entries.

Smallest Seen and $Av^r(123, 112)$

Words that avoid 123 and 112 are uniquely determined by their smallest seen entries.

Lemma: If $\pi, \pi' \in Av^r{}_m(123,112)$ have the same smallest seen entries, then $\pi = \pi'$.

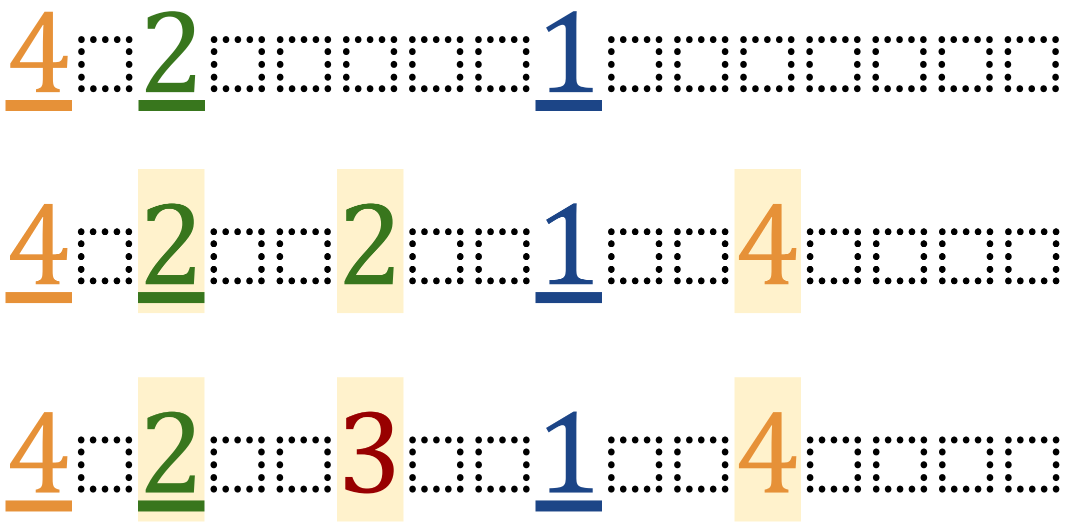
Proof: We claim that the non-smallest seen symbols are in *non-increasing order.*
Otherwise, we can prove that π will be forced to contain a 123 or 112 pattern.

$\underline{3}333\underline{2}222\underline{1}111$

Completing a word in $Av^4{}_3(123, 112)$ from its smallest seen entries.

$\underline{1}444333322211$

Completing a word in $Av^3{}_4(123, 112)$ from its smallest seen entries.



Completing a word $\pi \in Av^4_4(123, 112)$ from its <u>smallest seen</u> entries (aka left-to-right minima).

If the remaining symbols are not in non-increasing order, then there is an increase (e.g., 24 or 34).

- If the smaller symbol in the increasing pair has previously been seen, then π contains 112.
- If the smaller symbol in the increasing pair has not previously been seen, then π contains 123.

4□2□□□□□□□1□□□□□□□

44244333132221 11

Completing a word $\pi \in \mathrm{Av}^4_4(123, 112)$ from its smallest seen entries (aka left-to-right minima).
If the remaining symbols are not in non-increasing order, then there is an increase (e.g., 24 or 34).

- If the smaller symbol in the increasing pair has previously been seen, then π contains 112.
- If the smaller symbol in the increasing pair has not previously been seen, then π contains 123.

Mapping f_1 from r-Regular Words to Binary Strings

Define a mapping $f_1 : [m^r] \to \mathbb{B}(n+m)$ for $n = mr$ that works from from left-to-right as follows.

- Map smallest seen entries to $1^{s+1}0$, where s is counts the newly skipped symbols from $m, \ldots, 1$.
- Map every other entry to 0.

skipped $4, 3$ so $s=2$ | no new skips so $s=0$ | $s+1=3$ copies of 1 | $s+1=1$ copies of 1

$f_1(\underline{3}3\underline{2}33222\underline{1}111) = \underline{10}0\underline{10}000000\underline{10}000$ $f_1(\underline{2}443\underline{1}321) = \underline{1110}00000\underline{10}000$

Easy case: no skipped symbols.
The smallest seen are in the order $3, 2, 1$.

Skipped smallest symbols.
The smallest seen are in the order ~~$4, 3,$~~ $2, 1$.
So the number of new skips are 2 then 0.

Lemma: The restriction $f_1 \mid Av^r{}_m(123,112)$ is one-to-one to $(r+1)$-ary Dyck words $\mathbb{D}^{r+1}(m)$.

Proof: If $\pi \in Av^r{}_m(123,112)$, then $f_1(\pi) \in \mathbb{D}^{r+1}(m)$ because it contains m copies of 1, and each member of $[m]$ creates one copy of 1 and $1+(r-1) = r$ later copies of 0. The restriction is one-to-one from the previous uniqueness lemma, and that $f_1(\pi) \neq f_1(\pi')$ when π and π' have different smallest seen entries.

Inverse of f_1 from Dyck Words to $Av^r(123, 112)$

Scan the Dyck word from left to right.

- Map maximal blocks of the form $1^{i+1}0$ to the smallest seen symbol skipping over i symbols.
- Map other copies of 0 to largest remaining unexhausted symbol.

$\underline{110}0\underline{10}000000000000\underline{110}00000$

Member of $\mathbb{D}^{3+1}(5)$.

$\underline{4}5\underline{3}555444333322\underline{1}22111$

Member of $Av^3{}_5(123, 112)$.

The result is in $[m^r]$ and avoids 123 and 112 by the previous discussion.

Case ① Bijection $f_1 : Av^2_3(123, 112) \to \mathbb{D}^3(3)$ $r = 2$ and $m = 3$

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \rightarrow \mathbb{D}^3(3)$

r = 2 and m = 3

First left-to-right minima

- 3 → 10 set low=3
- 2 → 110 set low=2
- 1 → 1110 set low=1

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \rightarrow \mathbb{D}^3(3)$

r = 2 and m = 3

First left-to-right minima

3 → 10 set low=3
2 → 110 set low=2
1 → 1110 set low=1

Non-left-to-right minima

Fill with zeroes.

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \to \mathbb{D}^3(3)$

$r = 2$ and $m = 3$

First left-to-right minima

3 → 10 set low=3
2 → 110 set low=2
1 → 1110 set low=1

Non-left-to-right minima

Fill with zeroes.

Second left-to-right minima

if low=3 2 → 10 set low=2
if low=3 1 → 110 set low=1
if low=2 1 → 10 set low=1

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \to \mathbb{D}^3(3)$

r = 2 and m = 3

First left-to-right minima

3 → 10	set low=3
2 → 110	set low=2
1 → 1110	set low=1

Non-left-to-right minima

Fill with zeroes.

Second left-to-right minima

if low=3	2 → 10	set low=2
if low=3	1 → 110	set low=1
if low=2	1 → 10	set low=1

Non-left-to-right minima

Fill with zeroes.

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \to \mathbb{D}^3(3)$

r = 2 and m = 3

First left-to-right minima

3 → 10 set low=3
2 → 110 set low=2
1 → 1110 set low=1

Non-left-to-right minima

Fill with zeroes.

Second left-to-right minima

if low=3 2 → 10 set low=2
if low=3 1 → 110 set low=1
if low=2 1 → 10 set low=1

Non-left-to-right minima

Fill with zeroes.

Third left-to-right minima

if low=2 1 → 10 set low=1

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1 : Av^2_3(123, 112) \to \mathbb{D}^3(3)$

r = 2 and m = 3

First left-to-right minima

	3 → 10	set low=3
	2 → 110	set low=2
	1 → 1110	set low=1

Non-left-to-right minima

Fill with zeroes.

Second left-to-right minima

if low=3	2 → 10	set low=2
if low=3	1 → 110	set low=1
if low=2	1 → 10	set low=1

Non-left-to-right minima

Fill with zeroes.

Third left-to-right minima

if low=2	1 → 10	set low=1

Non-left-to-right minima

Fill with zeroes.

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

10 → 3 set low=3

110 → 2 set low=2

1110 → 1 set low=1

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

10 → 3 set low=3

110 → 2 set low=2

1110 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

- 10 → 3 set low=3
- 110 → 2 set low=2
- 1110 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Second maximal 1*0 block

- if low=3 10 → 2 set low=2
- if low=3 110 → 1 set low=1
- if low=2 10 → 1 set low=1

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

10 → 3 set low=3
110 → 2 set low=2
1110 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Second maximal 1*0 block

if low=3 10 → 2 set low=2
if low=3 110 → 1 set low=1
if low=2 10 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$ r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

10 → 3 set low=3
110 → 2 set low=2
1110 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Second maximal 1*0 block

if low=3 10 → 2 set low=2
if low=3 110 → 1 set low=1
if low=2 10 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Third maximal 1*0 block

10 → 1 set low=1

Case ① Bijection

$f_1^{-1} : Av^2_3(123, 112) \leftarrow \mathbb{D}^3(3)$

r = 2 and m = 3

332211	100100100
332121	100101000
331221	100110000
323211	101000100
323121	101001000
321321	101010000
313221	101100000
233211	110000100
233121	110001000
231321	110010000
213321	110100000
133221	111000000

First maximal 1*0 block

- 10 → 3 set low=3
- 110 → 2 set low=2
- 1110 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

Second maximal 1*0 block

- if low=3 10 → 2 set low=2
- if low=3 110 → 1 set low=1
- if low=2 10 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.

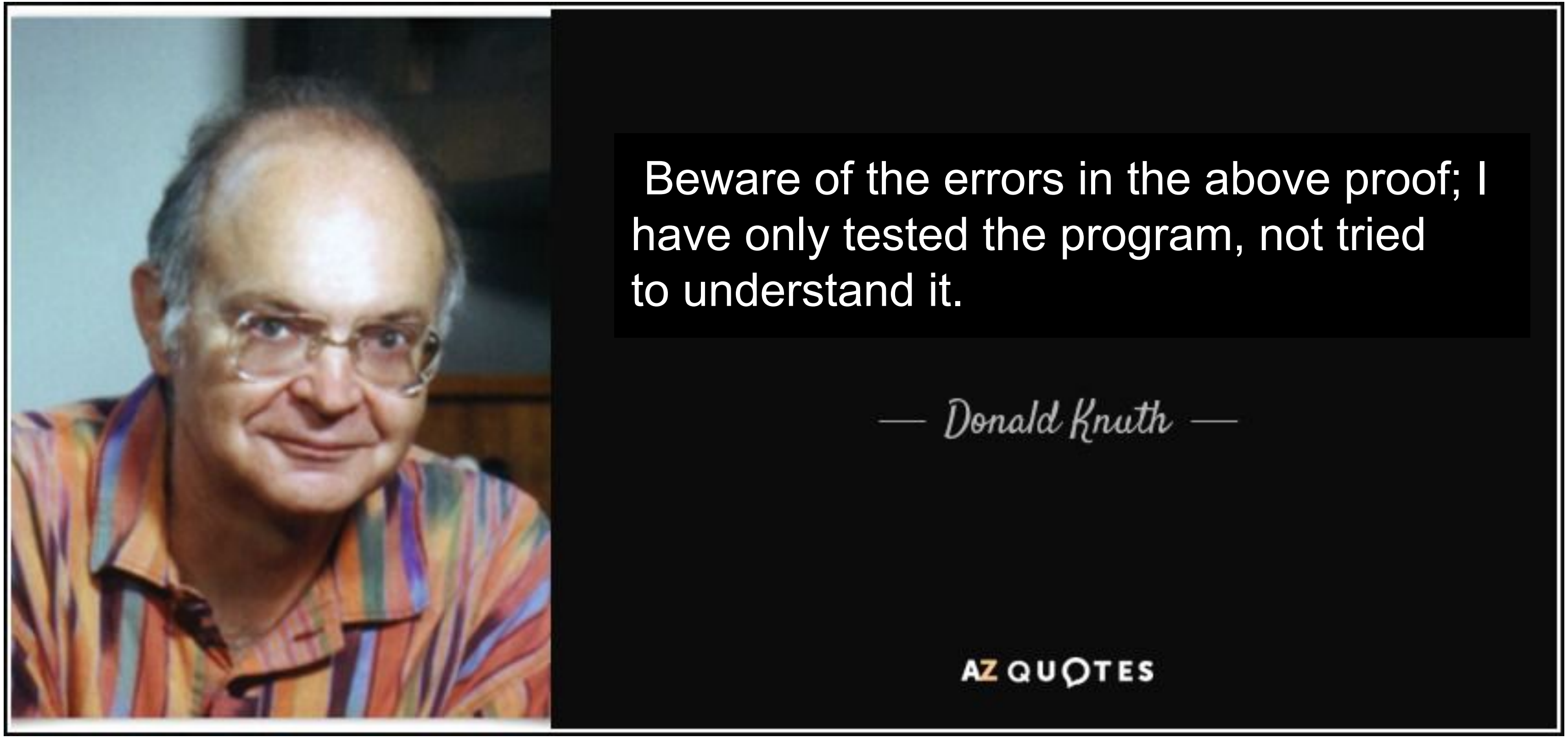
Third maximal 1*0 block

- 10 → 1 set low=1

Extra 0s

Fill with largest remaining symbols.


Beware of bugs in the above code; I have only proved it correct, not tried it.
— Donald Knuth —
AZ QUOTES


Beware of the errors in the above proof; I have only tested the program, not tried to understand it.
— Donald Knuth —
AZ QUOTES

```
# Input:  perm ∈ Av^r_m(123,112)
# Output: dyck ∈ D^{r+1}(m)
def f1(perm, m, r):
  low = m+1
  dyck = []
  for symbol in perm:
    if symbol < low:
      diff = low-symbol
      dyck += [1]*diff
      low = symbol
    dyck += [0]
  return dyck
```

```
# Input:  dyck ∈ D^{r+1}(m)
# Output: perm ∈ Av^r_m(123,112)
def f1inv(dyck, m, k):
  perm = () # Built one symbol at a time.
  low = m+1 # Left-to-right minima in perm.
  run = 0   # Current run of 1s in Dyck.
  Q = []    # Symbols to add to perm.
  for bit in dyck:
    run += bit
    if run == 0:
      perm = perm + (Q.pop(),)
    elif bit == 0:
      Q[:0] = [s for s in range(low-run,low)
                 for _ in range(k-1)][1:]
      low -= run
      perm = perm + (low,)
      run = 0
  return perm
```

Python code for computing f_1 and its inverse. The functions run in linear-time.

```python
def containsPattern(perm, pattern, forceLast=False):
    n = len(perm)
    m = len(pattern)
    bits = [1]*((m-1) if forceLast else m) + [0]*(n-m)
    for subCombo in coolMultiperms(bits):
        combo = tuple(subCombo) + (1,) if forceLast else tuple(subCombo)
        subperm = tuple(perm[i] for i in range(n) if combo[i] == 1)
        suborder = tuple(sorted(set(subperm)).index(x)+1 for x in subperm)
        if suborder == tuple(pattern):
            return True
    return False

def multipermsAvoidingPatterns(multiset, patterns, prefix = ()):
    if len(multiset) == 0:
        yield prefix
        return
    for s in set(multiset):
        for pattern in patterns:
            if containsPattern(prefix + (s,), pattern, True):
                return
    for s in set(multiset):
        index = multiset.index(s)
        multisetMinus = multiset[:index] + multiset[index+1:]
        prefixPlus = prefix + (s,)
        yield from multipermsAvoidingPatterns(multisetMinus, patterns, prefixPlus)
```

Only check patterns that include the most recently added symbol.

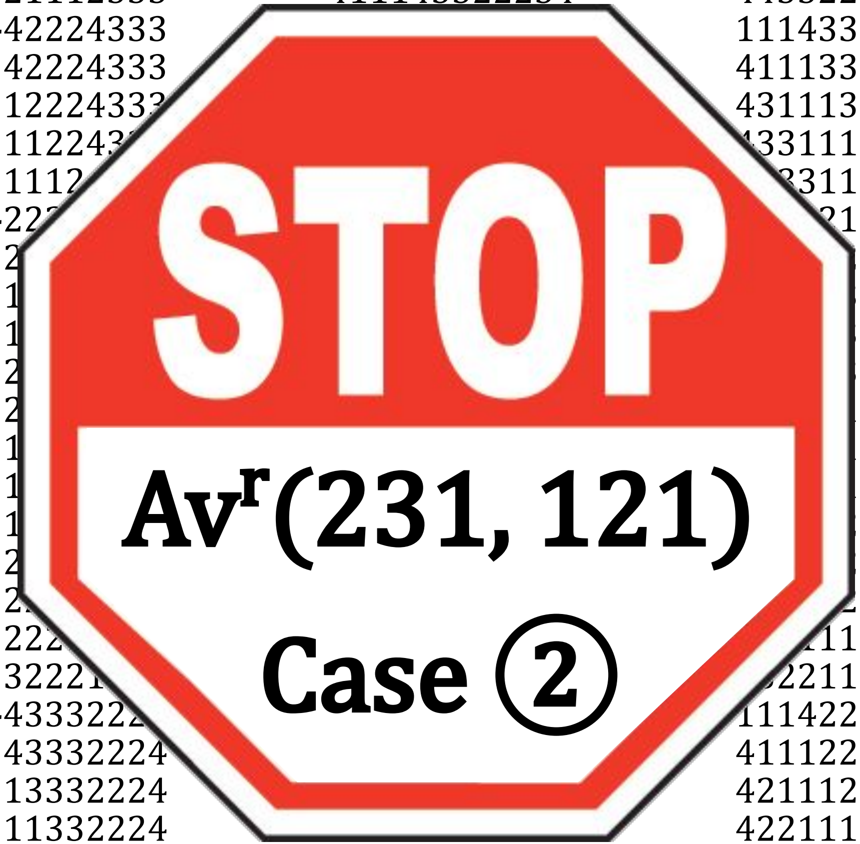
When backtracking make sure that every remaining symbol can be added.

Python code for generating all multiset permutations that avoid a list of patterns.

111444333222
411144333222
441114333222
444111333222
444311133222
444331113222
444333111222
444333211122
444333221112
111444332223
411144332223
441114332223
444111332223
444311132223
444331112223
444332111223
444332211123
111444322233
411144322233
441114322233
444111322233
444311122233
444321112233
444322111233
111444222333
411144222333
441114222333
444111222333

444211122333
444221112333
111442224333
411142224333
44111222433
4421112243
44221112
111422
41112
42111
42211
11122
21112
22111
22211
42221
44222
44422
4443222
444332221
1114433322
411143332224
441113332224
443111332224
443311132224
443331112224
443332111224
443332211124

111443322234
411143322234



221112443334
222111443334
422211143334
442221113334

443222111334
443322211134
111433322244
411133322244
431113322244
433111322244
3311122244
21112244
111244
222344
222344
222344
222344
122344
112344
223344
223344
223344
11223344
2211123344
111422233344
411122233344
421112233344
422111233344
111222433344
211122433344
221112433344
222111433344

422211133344
432221113344
433222111344
111333222444
311133222444
331113222444
333111222444
333211122444
333221112444
111332223444
311132223444
331112223444
332111223444
332211123444
111322233444
311122233444
321112233444
322111233444
111222333444
211122333444
221112333444
222111333444
322211133444
332221113444
333222111444
433322211144
443332221114
444333222111

First Occurrences

The *first occurrences* of a word are its the values and their first (i.e., leftmost) position in the word. In other words, (v, k) is a first occurrence if $p_1\, p_2 \cdots p_{k\text{-}1}$ does not contain v and $p_k = v$.

The *i^{th} first occurrence* refers to the i^{th} first occurrence when the word is read from left-to-right. In other words, we order the first occurrences by positions rather than values.

$p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12}$

443111322234

4□31□□□2□□□□

Example of first occurrences: (4,1), (3,3), (1,4) (2,8).

$p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12}$

111333222444

1□□3□□2□□4□□

Maximum positions of first occurrences in an r-regular word: 1, 4, 7, 10.

The first symbol is always the 1st first occurrence.

Remark: In an r-regular word, there are r first occurrences, and the position of i^{th} first occurrence is at most $r \cdot (i\text{-}1)+1$.

This remark is highly suggestive of a k-ary Dyck word for $k = r + 1$.

First Occurrences and Avoiding 231, 121

Words that avoid 231 and 121 are uniquely determined by their content and first occurrences.

Lemma: If $\pi, \pi' \in Av^r_m(231,121)$ have the same first occurrences, then $\pi = \pi'$.

Proof: Let $n = rm$ and $\pi = p_1\, p_2 \cdots p_{rm} \in Av^r_m(231,121)$. Consider p_d for $d = 1, 2, \ldots, n$.
If p_d is not a first occurrence, then we'll prove that it is completely determined by $p_1\, p_2 \cdots p_{d-1}$.
Let $\#_{k,d}$ be the number of copies of $k \in [m]$ before p_d. That is, $\#_{k,d} = |\{i : p_i = k \text{ for } 1 \leq i < d\}|$.
We claim that p_d is the *smallest possible symbol* (i.e., it is the smallest seen which isn't exhausted).
That is, $p_d = \min(k \in [m] \mid 0 < \#_{k,d} < r)$.

4□31□□□2□□□□	4□31□□□□2□□□	4□31□□□2□□□□
443111322234	□□□□13□□□□1□	□□□□□14□□□3□
Completing $\pi \in Av^3_4(231,121)$ from firsts.	Creating a 121.	Creating a 132.

Otherwise, there are two cases to consider for $p_d = x$.

1. If x is the second-smallest possible symbol, then a 121 pattern will be created.
2. If x is not the second-smallest possible symbol, then a 132 pattern will be created.

Mapping f_2 from r-Regular Words to Binary Strings

Define a mapping $f_2 : [m^r] \to \mathbb{B}(n+r)$ for $n = mr$ to transform the input from left-to-right as follows.

- Replace each first occurence of a symbol with 1, and every other occurrence of a symbol with 0.
- Insert an extra 0 for the last occurrence of each $v \in [m]$. But batch the insertions follows: Insert ℓ copies for the largest value of ℓ in which all $1, 2, \ldots, v\text{-}1, v+1, \ldots, v+\ell\text{-}1$ finish earlier. i.e., v-1, smaller values and ℓ-1 larger values have earlier last occurrences than v.

last 1 last 2 last 3 last 4 — 1 2 3 4

$$f_2(443111322234) = 1011000010000000$$

101100010000

Easy case: no batching of extra 0s.
The last occurrences are in the order $1, 2, \ldots, m$.

last 3 last 1 last 2 last 4 — 3 1 2 4

$$f_2(43321124) = 110110000000$$

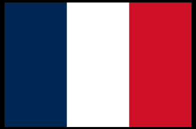
11011000

Batching of extra 0s.
The last occurrences are in the order 3,1,2,4.
Thus, 3's extra 0 is added after 2's extra 0.

Lemma: The restriction $f_2 \mid Av^r{}_m(231,121)$ is one-to-one to $(r+1)$-ary Dyck words $\mathbb{D}^{r+1}(m)$.

Proof: If $\pi \in Av^r{}_m(231,121)$, then $f_2(\pi)$ is a Dyck word by the first occurrence positions remark. The restriction is one-to-one from the previous uniqueness lemma, and that $f_2(\pi) \neq f_2(\pi')$ when π and π' have different first occurrences.

Aiguille (ay·**gwee**) a large, sharp peak. [French for "needle".]
peakhigh.co.za/mountain-terminology/

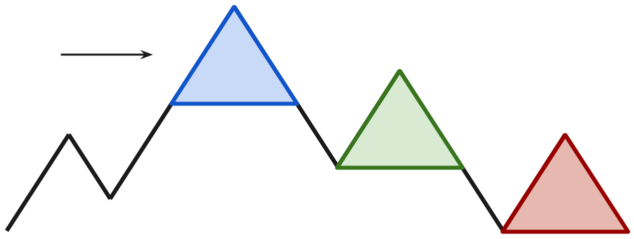


Aiguille Reductions and the Inverse of f_2

There are two natural notions of an "apex" in a k-ary Dyck word. (They are equivalent when $k=2$.)

- A *peak* is a substring of the form 10.
- An *aiguille* is a substring of the form 10^{k-1}. Or 10^r in an $(r+1)$-ary Dyck word.

Every non-empty k-ary Dyck word contains at least one aiguille.



10011000001000001000 $\xleftarrow{f_2}$ □□□□111□□□□□□□□

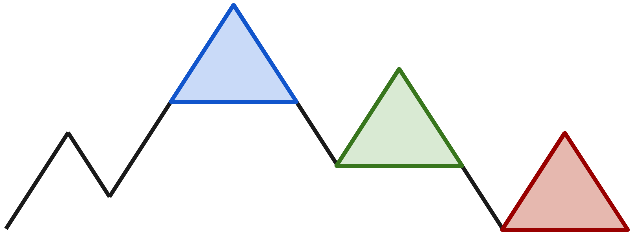
A 4-ary Dyck word, its Dyck path, and aiguilles.

The 3-ary word that maps to the Dyck word.

If $\pi \in Av^r(231,121)$, then the leftmost aiguille created by f_2 is from the 1^r substring in π.

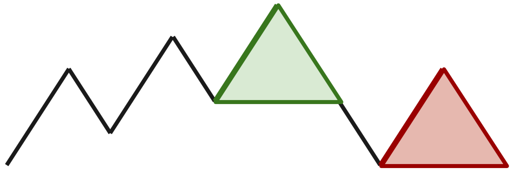
- The 1 is from a first occurrence. The next 0^{r-1} are not first occurrences.
- The last 0 is an extra. It must be obtained from the last copy of 1.

The inverse $f_2^{-1} : \mathbb{D}^{r+1}(m) \rightarrow Av^r_m(231,121)$ applies this recursively w/ successively larger symbols.



10011000001000001000 □□□□111□□□□□□□□

An example aiguille reduction.



1001001000001000

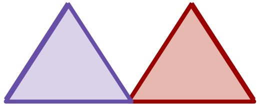
□□□□111□□222□□□

An example aiguille reduction.

- Note than new aiguilles can be created.

An example aiguille reduction.

- Note than new aiguilles can be created.

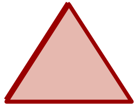


10001000

444311133222□□□

An example aiguille reduction.

- Note than new aiguilles can be created.



1000

444311133222555

An example aiguille reduction.

- Note than new aiguilles can be created.

Case ② Bijection

$f_2 : Av^2_3(231, 121) \leftrightarrow \mathbb{D}^3(3)$

$r = 2$ and $m = 3$

112233	100100100
113223	100110000
113322	100101000
211233	110000100
221133	101000100
311223	110010000
311322	110001000
321123	111000000
322113	110100000
331122	101001000
332112	101100000
332211	101010000

Case ② Bijection

$f_2 : Av^2_3(231, 121) \leftrightarrow \mathbb{D}^3(3)$

r = 2 and m = 3

112233	100100100
113223	100110000
113322	100101000
211233	110000100
221133	101000100
311223	110010000
311322	110001000
321123	111000000
322113	110100000
331122	101001000
332112	101100000
332211	101010000

Case ② Bijection

$f_2 : Av^2_3(231, 121) \leftrightarrow \mathbb{D}^3(3)$

$r = 2$ and $m = 3$

112233	100100100
113223	100110000
113322	100101000
211233	110000100
221133	101000100
311223	110010000
311322	110001000
321123	111000000
322113	110100000
331122	101001000
332112	101100000
332211	101010000

Case ② Bijection $f_2 : Av^2_3(231, 121) \leftrightarrow \mathbb{D}^3(3)$ $r = 2$ and $m = 3$

112233	100100100
113223	100110000
113322	100101000
211233	110000100
221133	101000100
311223	110010000
311322	110001000
321123	111000000
322113	110100000
331122	101001000
332112	101100000
332211	101010000

Additional Observations

These are conjectures that are "OEIS proven" experimentally!
In other words, each sequence matches *at most one* entry in the OEIS.

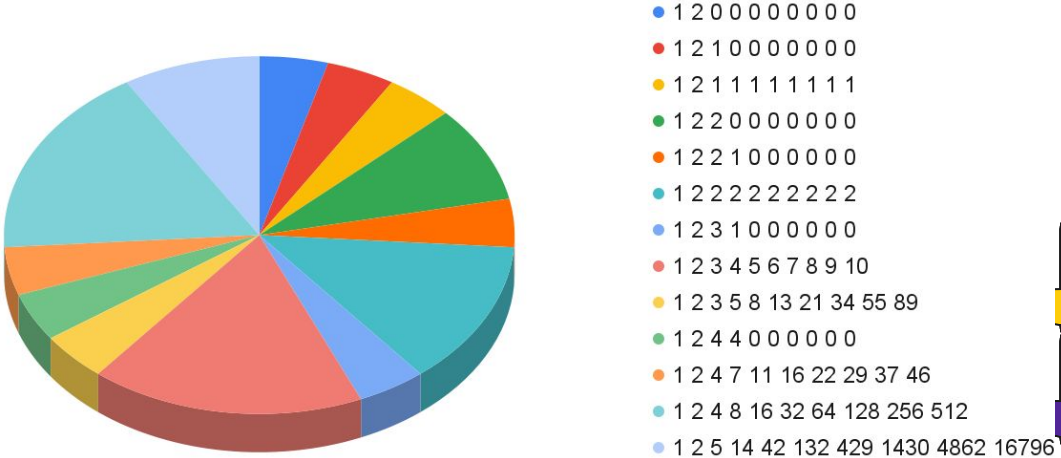

Exploring the Space

Subsets of Base Patterns (Alpha only)

Suppose that we have a set of base patterns. For example, standard length 3 patterns are below.

$$\mathrm{A} = \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$$

We want to run experiments on *all subsets* of some base patterns, simultaneously.
For the above set A, there are 18 non-empty subsets, up to isomorphisms of the square.
They give 13 different sequences for $\mathrm{r} = 1$ (i.e., permutations).



Fibonacci sequence from {123, 132, 213}

Catalan sequence from {123} and {132}

Some sequences are obtained by avoiding more than one subset of patterns from A.
$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, \ldots$ A000108 (Catalan) is obtained by two subsets.

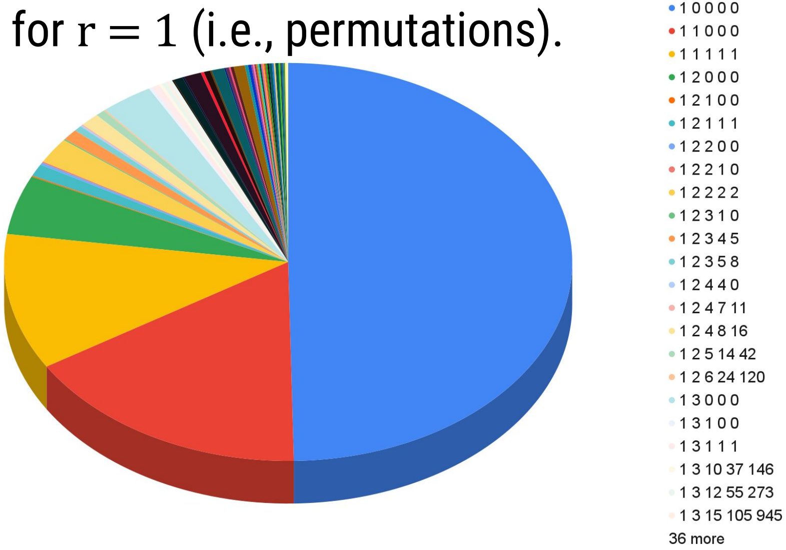
Classic sequences: $a(n) = n$, Fibonacci, Lazy Caterer $a(n) = (n+1)/2 + 1$, $a(n) = 2^n$, Catalan.

Subsets of Base Patterns (Alpha ∪ Beta)

In this talk we have been considering,

$$\Gamma = A \cup B = \{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\} \cup \{(1,1,2),(1,2,1),(1,2,2),(2,1,1),(2,1,2),(2,2,1)\}$$

There are 1103 non-isomorphic subsets (slightly more than $2^{12}/4$ by self-symmetries).
They give 59 different sequences for $r = 1$ (i.e., permutations).



Sequences from avoiding subsets of Γ-patterns over $\{1,1,2,2,3,3,\ldots,m,m\}$ for $m = 1,2,\ldots,5$.

Most of the sequences are uninteresting (e.g., 1 0 0 0 0 ...). Let's define this as $a(5) \leq 5$.

Alpha Patterns*

* in r-regular words

	123	132	123, 132	123, 231	132, 213	132, 231	132, 312
r = 1	A000108	A000108	A000079	A000124	A000079	A000079	A000079
	Catalan numbers	Catalan numbers	Powers of 2	Central polygonal numbers	Powers of 2	Powers of 2	Powers of 2
r = 2	A220097	A220097	A164908	A054567	A100191	A099856	A164908
	Avoid 123 in 2-regular	Avoid 123 in 2-regular	Cellular automata 14, 46, 142, 174	Ulam's spiral (west spoke)	Related to Pascal tetrahedron	Hankel transform 1,-18,0,0,...	Cellular automata 14, 46, 142, 174
r = 3	A266736	A266736					
	Avoid 123 in 3-regular	Avoid 123 in 3-regular					
r = 4	A266739	A266739					
	Avoid 123 in 4-regular	Avoid 123 in 4-regular					

Forbidding singletons and pairs from A = {(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)} in r-regular words. Restricted to “interesting” sequences.

	123, 132, 213	123, 132, 231	123, 132, 312	123, 231, 312	132, 213, 231	123,132,213,231
r = 1	A000108	A000027	A000027	A000027	A000027	
	Catalan numbers	Positive integers	Positive integers binomial(1n, 1)	Positive integers	Positive integers	
r = 2	A122365		A000384			
	2a(n-1) + 4a(n-2) - a(n-3) a(0) = 0 and a(1) = a(2) = 1		Hexagonal numbers binomial(2n, 2)			
r = 3			A006566			
			Dodecahedral numbers binomial(3n, 3)			
r = 4			A060541			
			binomial(4n, 4)			

Forbidding triplets and quartets from $A = \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$ in r-regular words. Restricted to “interesting” sequences.

Beta Patterns

	112	121	112, 121	112, 212	121, 212
r = 1	A000142	A000142	A000142	A000142	A000142
	Factorial numbers	Factorial numbers	Factorial numbers	Factorial numbers	Factorial numbers
r = 2	A001147	A001147	A000012	A000108	A000142
	Stirling permutations (2n-1)!!	Stirling permutations (2n-1)!!	Ones	Catalan	Factorial numbers
r = 3	A007559	A007559	A000012	A000108	A000142
	Triple factorial (3n-2)!!!	Triple factorial (3n-2)!!!	Ones	Catalan	Factorial numbers
r = 4	A007696	A007696	A000012	A000108	A000142
	Quartic factorial (4n-3)!!!!	Quartic factorial (4n-3)!!!!	Ones	Catalan	Factorial numbers

Forbidding subsets of B = {(1,1,2), (1,2,1), (1,2,2), (2,1,1), (2,1,2), (2,2,1)} in r-regular words.


(α, β)-Pairs

Avoiding (α, β) Patterns

There are $6 \cdot 6/4 = 9$ unique (α, β) pairs with $\alpha \in \{123, 132, 213, 231, 312, 321\}$ and $\beta \in \{112, 121, 122, 211, 212, 221\}$.
Below are the integer sequences that are obtained by avoiding each of these pairs in k-regular words.

classes	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
representative	(123,112)	(123,121)	(123,211)	(132,112)	(132,121)	(132,122)	(132,211)	(132,212)	(132,221)
reverse	(321,211)	(321,121)	(321,112)	(231,211)	(231,121)	(231,221)	(231,112)	(231,212)	(231,122)
inverse	(321,221)	(321,212)	(321,122)	(312,221)	(312,212)	(213,112)	(312,122)	(312,121)	(312,112)
both	(123,122)	(123,212)	(123,221)	(213,122)	(213,212)	(312,211)	(213,221)	(213,121)	(213,211)
k=2	A000108	A000108	A000108	A000108	A000108	A000108	A000108	A000108	A000108
k=3	A001764	A109081	trivial	A001333	A001764	A001764	A005408	A109081	A001333
k=4	A002293	A161797	trivial	A003688	A002293	A002293	A016777	A161797	A048654
k=5	A002294	A321798	trivial	A015448	A002294	A002294	A016813	A321798	A048655
k=6	A002295	A321799	trivial	A015449	A002295	A002295	A016861	A321799	A048693
k=7	A002296	new	trivial	A015451	A002296	A002296	A016921	new	A048694
k=8	A007556	new	trivial	A015453	A007556	A007556	A016993	new	A048695
	k-ary Catalan	Formula 1	1, k, 0, 0, ...	Generalized Fibonacci	k-ary Catalan	k-ary Catalan	Formula 2	Formula 1	Generalized Pellian

$$\sum_{i=0,1,\ldots,n} \frac{C(n,i)}{(n-i+1)} \cdot C(n+(k-2) \cdot i-1, n-i)$$

$$(k-1) \cdot n + 1$$

All experiments result in <u>at most one</u> OEIS sequence for large enough n.

	123, 112	123, 121	123, 211	132, 112	132, 121	132, 122	132, 211	132, 212	132, 221
r = 1	A000108	A000108	A000108	A000108	A000108	A000108	A000108	A000108	A000108
	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers	Catalan numbers
r = 2	A001764	A109081		A001333	A001764	A001764	A005408	A109081	A001333
	3-Catalan numbers				3-Catalan numbers	3-Catalan numbers			
r = 3	A002293	A161797		A003688*	A002293	A002293	A016777	A161797	A048654
	4-Catalan numbers				4-Catalan numbers	4-Catalan numbers			
r = 4	A002294	A321798		A015448	A002294	A002294	A016813	A321798	A048655
	5-Catalan numbers				5-Catalan numbers	5-Catalan numbers			
r = 5	A002295	A321799		A015449	A002295	A002295	A016861	A321799	A048693
	6-Catalan numbers				6-Catalan numbers	6-Catalan numbers			
r = 6	A002296	new		A015451	A002296	A002296	A016921	new	A048694
	7-Catalan numbers				7-Catalan numbers	7-Catalan numbers			
r = 7	A007556	new		A015453	A007556	A007556	A016993	new	A048695
	8-Catalan numbers				8-Catalan numbers	8-Catalan numbers			

Other $\Gamma = A \cup B$ Subsets?

(We'll look at one subset here)

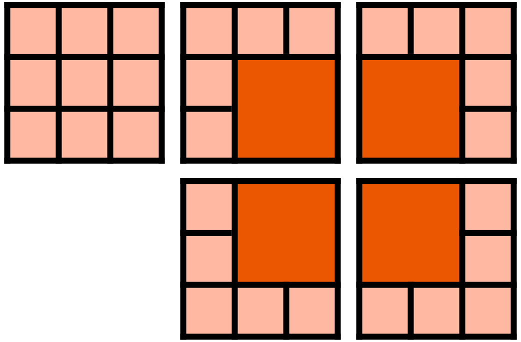
Ernst Jacobsthal

Jacobsthal Sequence

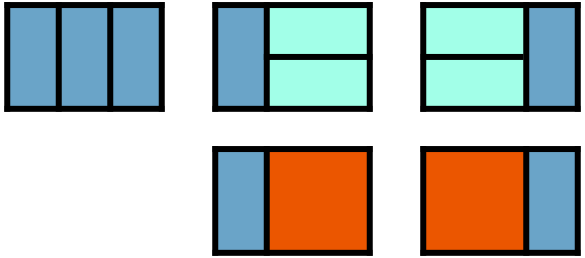
OEIS A001045: 0, 1, 1, 3, 5, 11, 21, 43, 85, 171, 341, 683, 1365, 2731, 5461, 10923, 21845, …

$$a(n) = a(n\text{-}1) + 2 \cdot a(n\text{-}2) \text{ with } a(0) = 0, a(1) = 1.$$

Jacobsthal Combinatorial Objects (n = 4)



Tile 3-by-(n-1) grid with 1-by-1 and 2-by-2 squares tiles.

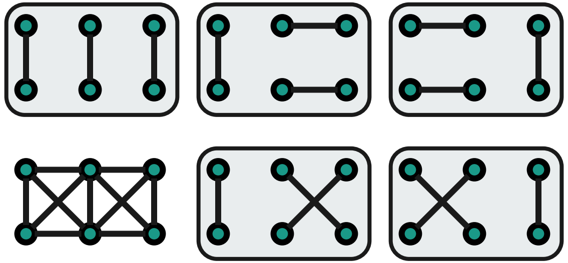


Tiling 2-by-(n-1) grid with dominoes (1-by-2 / 2-by-1) and 2-by-2 tiles.

000 010 100

011 110

Binary strings of length n-1 from (0 U 10 U 11)*.



Perfect matchings of a 2-by-n grid graph with added diagonal edges.

332211 322311 223311

332112 331122

New: $Av^2(121,123,132,213)$

Thank You!
Merci!