

Sorting Pattern-Avoiding Permutations via 0–1 Matrices Forbidding Product Patterns*

Parinya Chalermsook
Aalto University

Seth Pettie
University of Michigan

Sorrachai Yingchareonthawornchai
Aalto University and
Simons Institute for the Theory of Computing, UC Berkeley

Abstract

We consider the problem of comparison-sorting an n -permutation S that *avoids* some k -permutation π . Chalermsook, Goswami, Kozma, Mehlhorn, and Saranurak [CGK⁺15b] prove that when S is sorted by inserting the elements into the GREEDYFUTURE [DHI⁺09] binary search tree, the running time is linear in the *extremal function* $\text{Ex}(P_\pi \otimes (\cdot, \cdot), n)$. This is the maximum number of 1s in an $n \times n$ 0–1 matrix avoiding $P_\pi \otimes (\cdot, \cdot)$, where P_π is the $k \times k$ permutation matrix of π , and $P_\pi \otimes (\cdot, \cdot)$ is the $2k \times 3k$ Kronecker product of P_π and the “hat” pattern $(\cdot, \cdot)$. The same time bound can be achieved by sorting S with Kozma and Saranurak’s SMOOTHHEAP [KS20].

Applying off-the-shelf results on the extremal functions of 0–1 matrices, it was known that

$$\text{Ex}(P_\pi \otimes (\cdot, \cdot), n) = \begin{cases} \Omega(n\alpha(n)), \\ O\left(n \cdot 2^{(\alpha(n))^{3k/2 - O(1)}}\right), \end{cases}$$

where $\alpha(n)$ is the inverse-Ackermann function. In this paper we give *nearly tight* upper and lower bounds on the density of $P_\pi \otimes (\cdot, \cdot)$ -free matrices in terms of “ n ”, and improve the dependence on “ k ” from doubly exponential to singly exponential.

$$\text{Ex}(P_\pi \otimes (\cdot, \cdot), n) = \begin{cases} \Omega\left(n \cdot 2^{\alpha(n)}\right), & \text{for most } \pi, \\ O\left(n \cdot 2^{O(k^2) + (1+o(1))\alpha(n)}\right), & \text{for all } \pi. \end{cases}$$

As a consequence, sorting π -free sequences can be performed in $O(n2^{(1+o(1))\alpha(n)})$ time. For many corollaries of the dynamic optimality conjecture, the best analysis uses forbidden 0–1 matrix theory. Our analysis may be useful in analyzing other classes of access sequences on binary search trees.

1 Introduction

The problem of sorting restricted classes of permutations has been studied for decades. Knuth [Knu73] observed that the class of permutations sortable by a stack is precisely the set of $(2, 3, 1)$ -avoiding permutations; see [Tar72, BGH⁺10, MSS19, HI01, EG17a, EG17b, FP08, AB15, AMR02] and Bóna’s survey [Bón02] for models of restricted sorting devices. In general, an n -permutation S *avoids* a k -permutation π if there do not exist indices $i_1 < \dots < i_k$ for which

$$\forall p, q \in [k]. S(i_p) < S(i_q) \iff \pi(p) < \pi(q).$$

In this paper we consider the algorithmic problem of comparison-sorting a π -avoiding S .

*Supported by NSF grants CCF-1815316 and CCF-2221980 and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 759557.

Decision Tree Complexity. Fredman [Fre76] observed that if S is known to be selected from a permutation set Γ , that S can be sorted with $O(n + \log |\Gamma|)$ comparisons. The *Stanley-Wilf conjecture* (see Bóna [Bón22]) states that if Γ_π is the set of all π -avoiding permutations, that $|\Gamma_\pi| \leq (c(\pi))^n$, for some constant $c(\pi)$. This conjecture was reduced to the *Füredi-Hajnal conjecture* [FH92] by Klazar [Kla00] and both conjectures were proved by Marcus and Tardos [MT04]. Together with Fredman [Fre76], this implies that the decision-tree complexity of sorting S is $O(n \log c(\pi)) = O_k(n)$. Subsequent work has attempted to pin down the leading constant [Kla00, MT04, Cib09, Fox13, CK17]. Fox [Fox13] proved that¹

$$n \log c(\pi) = \begin{cases} O(kn) & \text{For all } k\text{-permutations } \pi, \\ \Omega(k^{1/2}n) & \text{For some } k\text{-permutation } \pi, \\ \Omega((k/\log k)^{1/2}n) & \text{For almost all } k\text{-permutations } \pi. \end{cases}$$

Algorithmic Complexity. There are two natural ways to approach the *algorithmic* complexity of sorting a π -free S . The first is to use knowledge of π to structure the sorting process. This approach is sufficient to sort optimally in $O(n)$ time when $k = 3$ [Knu73, Art07], and has had limited success for some patterns with $k = 4$. Arthur [Art07] gave $O(n)$ -time sorting algorithms when $\pi \in \{(1, 2, 3, 4), (1, 2, 4, 3), (2, 1, 4, 3)\}$, and $O(n \log \log \log n)$ -time sorting algorithms when $\pi \in \{(1, 3, 2, 4), (1, 3, 4, 2), (1, 4, 2, 3), (1, 4, 3, 2)\}$. The *oblivious* approach to sorting S is to simply use a general-purpose sorting algorithm, but analyze its behavior when S happens to be π -free. This is the approach taken by Chalermsook, Goswami, Kozma, Mehlhorn, and Saranurak [CGK⁺15b], Kozma and Saranurak [KS20], and by our paper. Consider these two general-purpose sorting algorithms:

BST Sort. Fix some dynamic binary search tree (BST) algorithm $\mathcal{T}$. Beginning from an empty BST, insert the elements $S(1), \dots, S(n)$ in that order, reorganizing the tree between inserts as $\mathcal{T}$ dictates. The number of comparisons is the sum of depths of $(S(i))_{1 \leq i \leq n}$ at the time of their insertion; the *time* is linear in the number of comparisons and that needed to reorganize the tree via rotations.

Heap Sort. Fix some heap data structure $\mathcal{H}$. Insert the elements $S(1), \dots, S(n)$ into the heap in that order, then perform n Delete-Min operations, thereby sorting the sequence.

Chalermsook et al. [CGK⁺15b] analyzed the performance of BST Sort when $\mathcal{T}$ is GREEDYFUTURE [DHI⁺09], an online BST that is $O(1)$ -competitive with the natural offline GREEDY algorithm [Luc88, Mun00]. Define A_S to be the $n \times n$ 0–1 permutation matrix where $A_S(i, S(i)) = 1$. If S avoids a k -permutation π , then A_S is P_π -free, where $P_\pi(i, \pi(i)) = 1$. Define $A_{\text{GREEDY}(S)}(i, j) = 1$ iff the element with rank j is touched by the insertion of $S(i)$. Chalermsook et al. [CGK⁺15b] proved that any occurrence of the “hat” pattern $(\cdot, \cdot)$ in $A_{\text{GREEDY}(S)}$ contains, within its bounding box, an input point of A_S , and as a consequence, $A_{\text{GREEDY}(S)}$ avoids $Q = P_\pi \otimes (\cdot, \cdot)$, where $\otimes$ is the Kronecker product, i.e., each 1 of P_π is replaced by $(\cdot, \cdot)$. (Following convention, 0–1 matrices are depicted with blanks for 0s and bullets for 1s. See Section 2 for explicit definitions regarding 0–1 matrices.) For example, if $\pi = (1, 3, 2, 4)$, ordering rows from bottom to top:

$$P_\pi = \begin{pmatrix} & & & \bullet \\ & \bullet & & \\ & & \bullet & \\ \bullet & & & \end{pmatrix} \quad Q = P_\pi \otimes (\cdot, \cdot) = \begin{pmatrix} & & & \bullet & \bullet \\ & & & \bullet & \bullet \\ & \bullet & \bullet & & \\ & & \bullet & \bullet & \\ \bullet & \bullet & & & \end{pmatrix}$$

If X is a fixed 0–1 pattern matrix, define $\text{Ex}(X, n)$ be the maximum number of 1s in an $n \times n$ matrix that avoids X . Thus, the running time of [CGK⁺15b] can be bounded in terms of the quantity $\text{Ex}(Q, n)$ without knowing exactly what it is.

THEOREM 1.1. (CHALERMSOOK, GOSWAMI, MEHLHORN, KOZMA, AND SARANURAK [CGK⁺15B]) *If S is π -free, BST Sort using GREEDYFUTURE sorts S in $O(\text{Ex}(Q, n))$ time, where $Q = P_\pi \otimes (\cdot, \cdot)$.*

¹The manuscript [Fox13] only gives an $\Omega(k^{1/4}n)$ lower bound on the decision tree complexity of sorting a π -free S . The $\Omega(k^{1/2}n)$ lower bound is unpublished.

Observe that Q is a $2k \times 3k$ *light* pattern: it contains exactly one 1 per column. There is a well known connection between light patterns and generalized Davenport-Schinzel sequences [Kla92, FH92, Kes09, Pet11b, Pet15b]. Applying a simplifying transformation that collapses the first two rows [FH92, Thm. 2.2] and then [Pet15b, Thm. 1.3], we have the following general upper bound, where $\alpha(n)$ is the inverse-Ackermann function.

$$(1.1) \quad \text{Ex}(Q, n) \leq \begin{cases} 2n\alpha(n) + O(n) & k = 2 \\ n \cdot 2^{(1+o(1))\alpha^t(n)/t!} & k \text{ odd, } t = (3k-5)/2 \\ n \cdot (\alpha(n))^{(1+o(1))\alpha^t(n)/t!} & k \text{ even, } t = (3k-6)/2 \end{cases}$$

Thus, by Theorem 1.1, GREEDYFUTURE sorts S in $O(n \cdot 2^{\alpha(n)^{3k/2-O(1)}})$ time. On the lower bound side, we know that $\text{Ex}(Q, n) = \Omega(n\alpha(n))$ as every Q contains one of the two patterns shown below, which are associated with order-3 Davenport-Schinzel sequences [HS86, FH92].

$$\begin{pmatrix} \cdot & & \\ & \cdot & \\ & & \cdot \end{pmatrix} \quad \begin{pmatrix} & & \cdot \\ \cdot & & \\ & \cdot & \end{pmatrix}$$

The GREEDY algorithm is theoretically attractive, but cumbersome to implement online as GREEDYFUTURE [DHI⁺09]. Kozma and Saranurak [KS20] introduced a new heap data structure called a SMOOTHHEAP, and proved **Heap Sort** with SMOOTHHEAP is *equivalent* to **BST Sort** with GREEDY. Moreover, SMOOTHHEAP is “naturally” an online algorithm, and is easier to implement than GREEDYFUTURE. One can define an $n \times n$ 0–1 matrix $A_{\text{SMOOTHHEAP}(S)}$ in the same way, where $A_{\text{SMOOTHHEAP}(S)}(i, j) = 1$ iff the i th Delete-Min touches the element with rank j . It is proved that $A_{\text{SMOOTHHEAP}}$ avoids a matrix equivalent to Q ².

THEOREM 1.2. (KOZMA AND SARANURAK [KS20]) *If S is π -free, **Heap Sort** using GREEDYFUTURE sorts S in $O(\text{Ex}(Q, n))$ time, where $Q = P_\pi \otimes (\cdot, \cdot)$.*

The main outstanding question is whether it is possible to sort in $O_k(n)$ time, and in particular, whether the GREEDY- or SMOOTHHEAP-based algorithms of [CGK⁺15b, KS20] already sort in time $O_k(n)$. It would also be interesting to give a non-trivial upper bound on the complexity of **BST Sort** with a SPLAY TREE [ST85], or **Heap Sort** with a PAIRING HEAP [FSST86].

1.1 New Results

1.1.1 Upper Bounds Our main result is a new upper bound on the extremal function of $P_\pi \otimes (\cdot, \cdot)$ -type matrices that has a much weaker dependence on k , which immediately gives better upper bounds on the complexity of sorting π -free sequences via [CGK⁺15b, KS20].

THEOREM 1.3. *Let P_π be the $k \times k$ permutation matrix of π and $Q = P_\pi \otimes (\cdot, \cdot)$ be a $2k \times 3k$ light matrix. Then*

$$\text{Ex}(Q, n) \leq n \cdot \left(2^{O(k^2)} + O(\alpha(n))^{3k-2} \right) 2^{\alpha(n)} = n \cdot 2^{O(k^2) + (1+o(1))\alpha(n)}.$$

COROLLARY 1.1. *If S is π -free, then **BST Sort** using GREEDYFUTURE and **Heap Sort** using the SMOOTHHEAP will sort S in $n \cdot 2^{O(k^2) + (1+o(1))\alpha(n)}$ time.*

One can view Corollary 1.1 as improving on the $n2^{\alpha(n)^{3k/2-O(1)}}$ bound of (1.1) in two ways. It is an asymptotic improvement in n as it brings the exponent of $\alpha(n)$ from $3k/2 - O(1)$ down to 1. However, even if one is tempted to consider $\alpha(n)$ to be a small constant, it also reduces the dependency on k from *doubly* exponential to merely *singly* exponential.

It is possible to improve the factor $2^{\alpha(n)}$ for a specific product pattern. For example,

THEOREM 1.4. *If I_k is the $k \times k$ identity matrix, then*

$$\text{Ex}(I_k \otimes (\cdot, \cdot), n) \leq 2(k-1)n\alpha(n) + O(kn).$$

²Strictly speaking the equivalence between GREEDY and SMOOTHHEAP swaps the roles of time and space. Sorting S with GREEDY is isomorphic to sorting S^T with SMOOTHHEAP, where S^T is the transpose permutation: $S(i) = j \Leftrightarrow S^T(j) = i$. Note that S^T avoids π^T . Since the extremal functions for Q and Q^T are identical on square matrices, we infer that the time to sort S^T is also $O(\text{Ex}(Q, n))$.

1.1.2 Lower Bounds When $k \geq 2$, all $Q = P_\pi \otimes (\cdot, \cdot)$ patterns contain $\begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}$ or its reflection, which is known to have extremal function $\text{Ex}(\begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}, n) = 2n\alpha(n) \pm O(n)$ [HS86, FH92, Niv10, Pet15a].

We prove that $\text{Ex}(P_\pi \otimes (\cdot, \cdot), n) = \Omega(n2^{\alpha(n)})$ whenever π contains $(3, 1, 2)$ or $(2, 1, 3)$, or equivalently, when P_π contains $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$ or $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$. Thus, Theorem 1.5 implies that the general upper bound of Theorem 1.3 can only be improved in the $\text{poly}(\alpha(n))$ factor.

THEOREM 1.5. $\text{Ex}(W, n) = \Theta(n2^{\alpha(n)})$, where

$$W = \begin{pmatrix} \cdot & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \cdot \end{pmatrix}.$$

1.2 Pattern-avoidance and the Dynamic Optimality Conjecture The original *dynamic optimality conjecture* [ST85] states that the (online) SPLAY BST is $O(1)$ -competitive with the optimum offline BST, for any sequence with length $\Omega(n)$. Today dynamic optimality usually refers to the conjecture that *there exists* an $O(1)$ -competitive BST, with GREEDY / GREEDYFUTURE [Luc88, Mun00, DHI⁺09] and SPLAY being the foremost candidates.

It is an open problem to prove $o(\log n)$ -competitiveness for SPLAY or GREEDY, though some *corollaries* of dynamic optimality have been proved [ST85, Tar85, Col00, CMSS00, IL16, CGJ⁺23, LT19]. Many corollaries of dynamic optimality can be characterized by forbidden patterns. For example, the offline optimum BST executes all of these sequences in linear time. The last item is the class of input sequences we consider in this paper. It subsumes all previous items, and was only recently shown to require $O_k(n)$ time by the optimum *offline* BST [BKO23].

Sequential. The sequential access sequence $S = (1, 2, \dots, n)$ avoids $(2, 1)$.

Deque. In a deletion-only deque sequence, $S(i)$ is either the minimum or maximum of $\{S(i), S(i+1), \dots, S(n)\}$. Deque sequences avoid $\{(213), (312)\}$. (In a deque, the accessed elements are also typically deleted from the tree [Sun92, Pet08].)

Preorder and Postorder. Let R be any BST over $\{1, \dots, n\}$ and S be a preorder (or postorder) traversal of R . Then S avoids (231) (or (312)). (The *Traversal Conjecture* of Sleator and Tarjan [ST85] concerned preorder sequences. If the accessed elements in a preorder sequence are moved to the root and *deleted*, yielding two trees, this corresponds with Lucas's definition of *Split*-sequences [Luc91].)

k -Increasing. S can be decomposed into $(k-1)$ increasing subsequences, or equivalently, S avoids $(k, \dots, 2, 1)$.

k -Recursively Decomposable. A permutation S is k -recursively decomposable if (i) the 1s of the corresponding permutation matrix A_S can be partitioned into k non-overlapping rectangles, and (ii) those rectangles are themselves k -recursively decomposable, where in the base case, any 1×1 matrix is k -recursively decomposable. These sequences avoid all *simple* $(k+1)$ -permutations³.

k -Permutation Avoiding. The access sequence is a permutation S , which avoids some k -permutation π .

Figure 1 shows the relationship between the classes of permutations, and Table 1 gives some known upper bounds on the performance of SPLAY and GREEDY. In particular, our new upper bound on $\text{Ex}(P_\pi \otimes (\cdot, \cdot), n)$ improves on the bounds for k -recursively decomposable sequences (when preprocessing is not allowed), and k -permutation avoiding sequences.

1.3 Organization In Section 2 we review forbidden 0–1 matrix terminology, and some key results. In Section 3 we prove Theorem 1.3, establishing the $n2^{(1+o(1))\alpha(n)}$ upper bound on $P \otimes (\cdot, \cdot)$ -type matrices. In Section 4 we prove Theorem 4.1's $\Omega(n2^{\alpha(n)})$ lower bound on W -free matrices. Section 5 presents some additional upper bounds, on $I_k \otimes (\cdot, \cdot)$ -free matrices (Theorem 1.4) and matrices avoiding W and its reflection. We conclude with some open problems in Section 6.

³A $(k+1)$ -permutation π is *simple* if there is no interval $I \subset \{1, \dots, k+1\}$ with $|I| \in [2, k]$ such that $\pi(I) \stackrel{\text{def}}{=} \{\pi(j) \mid j \in I\} = I$.

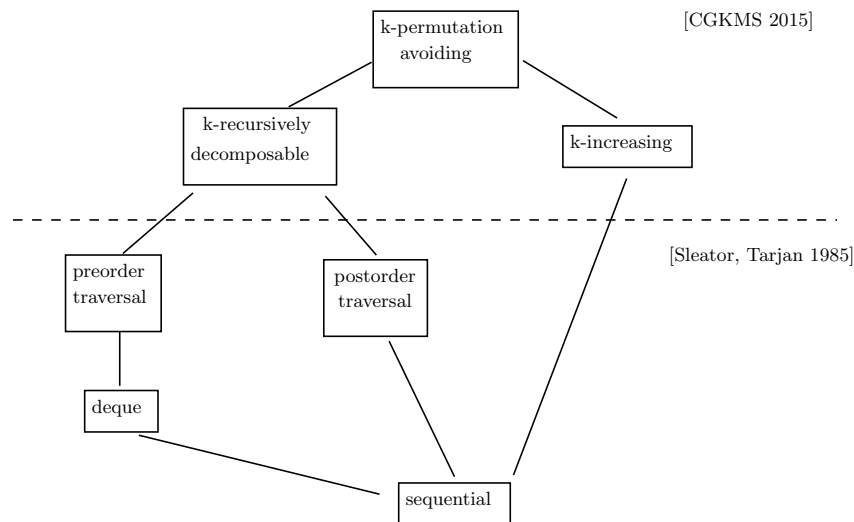


Figure 1: Relation between classes of search sequences. The upper class contains the lower ones.

Search Sequence	Forbidden Pattern	Splay	Greedy	Citation
Sequential	(21) -free	$O(n)$	$O(n)$	[Tar85]
Deque	$\{(213), (231)\}$ -free	$O(n\alpha^*(n))$	$O(n\alpha(n))$	[Pet08] [CGJ ⁺ 23]
Preorder	(231) -free	—	$O(n2^{\alpha(n)})$	[CGJ ⁺ 23]
Postorder	(312) -free	—	$O(n)$	[CGJ ⁺ 23]
k -Increasing	$(k, \dots, 2, 1)$ -free	—	$O(\min\{nk^2, nk\alpha(n)\})$	[CGJ ⁺ 23]
k -Recursively decomposable	avoids all <i>simple</i> $(k+1)$ -permutations	—	$O(n \log k)$ (<i>prepr. initial tree</i>)	[GG19]
k -Permutation avoiding	π -free	—	$O(\text{Ex}(P_\pi \otimes (\cdot, \cdot), n))$	[CGK ⁺ 15b]

Table 1: Upper Bounds on Structured Search Sequences

2 Preliminaries

Let $A \in \{0, 1\}^{n \times m}$ and $P \in \{0, 1\}^{k \times l}$. The *weight* of A , denoted as $\|A\|_1$, is the number of 1's in A . We say P is *contained* in A , written $P < A$ if there are row indices $r_1 < \dots < r_k$ and column indices $c_1 < \dots < c_l$ such that $P(i, j) = 1 \rightarrow A(r_i, c_j) = 1$. In other words, you can obtain P from A by deleting rows and columns, and flipping some 1s to 0. The extremal functions are defined as follows.

$$\begin{aligned} \text{Ex}(P, n, m) &= \max\{\|A\|_1 \mid A \in \{0, 1\}^{n \times m}, P \not< A\}, \\ \text{Ex}(P, n) &= \text{Ex}(P, n, n). \end{aligned}$$

If P is a $k \times k$ permutation matrix, it is known that both $\text{Ex}(P, n)$ and $\text{Ex}(P \otimes (\cdot), n)$ are $O_k(n)$, but we will be interested in the leading constants as well.

THEOREM 2.1. ([MT04], [GEN09], FOX [FOX13], [CK17], [GEN15], [GT17]) *Let P be any permutation matrix. Then there exists constants $C_k, C'_k \leq 2^{(4+o(1))k}$ such that*

$$\begin{aligned} \text{Ex}(P, n, m) &\leq C_k(n + m), \\ \text{Ex}(P \otimes (\cdot), n, m) &\leq C'_k(n + m). \end{aligned}$$

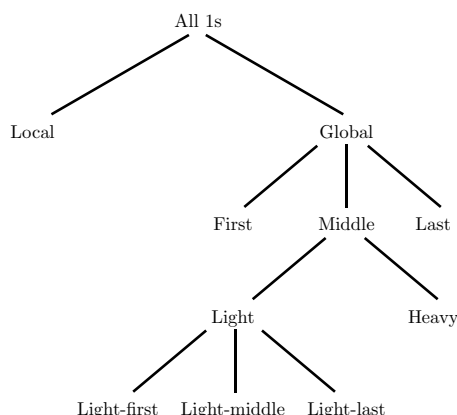
3 The Upper Bound

3.1 Establishing the General Recurrence Let P be a $k \times k$ permutation matrix and $Q = P \otimes (\cdot)$ be the $2k \times 3k$ forbidden pattern. Define $Q_{a,b}$ to be the $2k \times (3k - (a + b))$ matrix derived from Q by removing the first a and last b columns. For reasons that will become clear later, we must redefine the *contains* relation $<$ differently for the $Q_{a,b}$ matrices.

DEFINITION 3.1. *We will say that $Q_{a,b} < A$ if there are $2k$ rows $r_1 < \dots < r_{2k}$ and $3k - a - b$ columns $c_1 < \dots < c_{3k-a-b}$ such that*

- $Q_{a,b}(i, j) = 1$ implies $A(r_i, c_j) = 1$
- If $\forall j. Q_{a,b}(i, j) = 0$ then $\exists j'. A(r_i, j') \neq 0$. In other words, an all-0 row $Q_{a,b}(i, \cdot)$ cannot match an all-0 row of A . (Note that j' need not be in $\{c_1, \dots, c_{3k-a-b}\}$.)

Let A be an $n \times m$ $Q_{a,b}$ -free matrix with weight $\text{Ex}(Q_{a,b}, n, m)$. We will classify all 1s in A according to the following taxonomy, and bound the number of 1s in each class directly or inductively.



Partition A into *slabs* of B consecutive columns. A row is called *local* if it has a non-zero intersection with exactly one slab and *global* otherwise. The 1s in local/global rows are themselves local/global. Let n_i be the number of rows local to slab i and n^* be the number of global rows, so $n = n^* + \sum_i n_i$.

Suppose $A(r, c) = 1$ is a 1 appearing in a global row r and slab $s = \lceil c/B \rceil$. We classify this 1 as *first* if the intersection of row r and slabs $1, \dots, s-1$ are zero, *last* if the intersection of row r and slabs $s+1, \dots, \lceil m/B \rceil$ is zero, and *middle* otherwise.

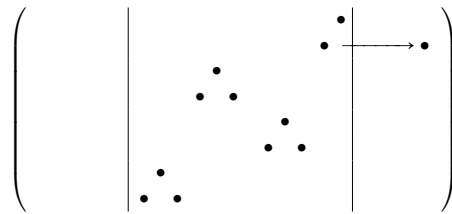


Figure 2: Vertical lines mark the boundary of some slab. If $Q_{0,1}$ appears in one slab of A_{first} , then there must be an occurrence of $Q = Q_{0,0}$ in A .

Since each slab is itself $Q_{a,b}$ -free, the total number of local 1s is at most

$$(3.2) \quad \sum_{i=1}^{\lceil m/B \rceil} \text{Ex}(Q_{a,b}, n_i, m_i),$$

where m_i is the number of columns in slab i , which is exactly B except perhaps the last slab. Similarly, if A_{first} and A_{last} are the matrices of first 1s and last 1s, then each slab of A_{first} is $Q_{a,b+1}$ free, and each slab of A_{last} is $Q_{a+1,b}$ -free; see Figure 2. Letting n_i^f (n_i^l) be the number of rows with first (last) 1s in slab i , we can upper bound first and last 1s as follows.

$$(3.3) \quad \begin{aligned} \|A_{\text{first}}\|_1 + \|A_{\text{last}}\|_1 &\leq \sum_{i=1}^{\lceil m/B \rceil} \left(\text{Ex}(Q_{a,b+1}, n_i^f, m_i) + \text{Ex}(Q_{a+1,b}, n_i^l, m_i) \right) \\ &\leq \text{Ex}(Q_{a,b+1}, n^*, m - m_{\lceil m/B \rceil}) + \text{Ex}(Q_{a+1,b}, n^*, m - m_1). \end{aligned}$$

In Eqn. (3.3) we use the *superadditivity* of Ex to simplify the expression. For any R , $\text{Ex}(R, n_1, m_1) + \text{Ex}(R, n_2, m_2) \leq \text{Ex}(R, n_1 + n_2, m_1 + m_2)$. Note that $\sum_i n_i^f = \sum_i n_i^l = n^*$ and that the first and last slabs contain no last 1s and first 1s, respectively.

Let A^* be the $n^* \times m$ matrix formed by the global rows and containing only the middle 1s. We partition the rows of A^* into horizontal slabs of G rows each, so the intersections of the horizontal and vertical slabs induce $G \times B$ blocks. Call a $G \times B$ block in A^* *heavy* if it contains a $(\cdot, \cdot)$, and *light* otherwise. The middle 1s inside heavy/light blocks are themselves called heavy/light. Let A_{heavy} and A_{light} be the $n^* \times m$ matrices containing heavy and light 1s, respectively. In a light block, the first 1 and last 1 of each row are called *light-first* and *light-last*, and all other 1s in the row are *light-middle*.

Define A_{heavy}^c to be the $n^*/G \times m/B$ matrix obtained by *contracting* each block in A_{heavy} to a single entry, i.e., non-zero blocks become 1 and all-zero blocks become 0. Because each heavy block contains a $(\cdot, \cdot)$, A_{heavy}^c is P -free, implying $\|A_{\text{heavy}}^c\|_1$ (the number of heavy blocks) is at most $\text{Ex}(P, n^*/G, m/B)$. Since each heavy block consists solely of middle 1s, each is $Q_{a+1,b+1}$ -free; see Figure 3. Thus,

$$(3.4) \quad \|A_{\text{heavy}}\|_1 \leq \text{Ex}(P, n^*/G, m/B) \cdot \text{Ex}(Q_{a+1,b+1}, G, B).$$

Let A_{light}^c be obtained by contracting the B columns in each slab of A_{light} to a single column. A_{light}^c inherits the $Q_{a,b}$ -freeness of A_{light} and A , so the contribution of light 1s in the light-first and light-last categories is at most

$$(3.5) \quad 2\|A_{\text{light}}^c\|_1 \leq 2\text{Ex}(Q_{a,b}, n^*, m/B).$$

What remains is to bound the light 1s in the light-middle category. Construct an $n^*/G \times m/B$ matrix A_{lightmid} by the following procedure, which is similar to that of [Gen09]. Assume the rows of A_{lightmid} are numbered from bottom to top. For each i independently, scan the blocks in slab i that contain light-middle 1s from bottom to top, setting $A_{\text{lightmid}}(\ell_0, i) = A_{\text{lightmid}}(\ell_1, i) = \dots = 1$ according to the following rules. See Figure 4.

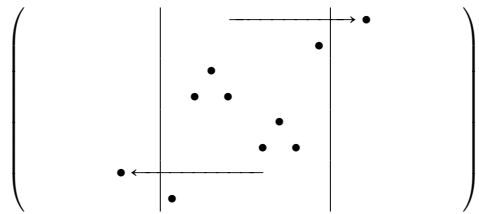


Figure 3: If an instance of $Q_{2,2}$ is contained in a single slab of middle 1s (e.g., A_{heavy} or A_{light}), then $Q_{1,1}$ must also appear in A . This inference relies on how *contains* is defined for $Q_{a,b}$ matrices in Definition 3.1. In particular, it is critical that all-zero rows of $Q_{2,2}$ must *not* be all-zero in the instance of middle 1s.

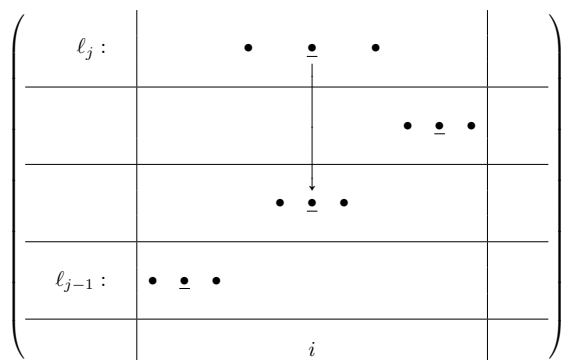


Figure 4: Vertical and horizontal lines mark block boundaries. Underlined 1s are light-middle 1s.

1. (ℓ_0, i) is the first block in slab i containing a light-middle 1.
2. $\ell_j > \ell_{j-1}$ is the first index such that some column in blocks $(\ell_{j-1}, i), \dots, (\ell_j, i)$ contains two light-middle 1s.

Call the interval of blocks $(\ell_{j-1}, i), \dots, (\ell_j - 1, i)$ in A_{light} (i.e., excluding (ℓ_j, i)) a *chunk*. By construction, the intersection of a column and a chunk can contain at most one light-middle 1. (Note that no light block contains two light-middle 1s in the same column, for otherwise it would contain a $(\cdot, \cdot)$ pattern and be classified as heavy.) We claim A_{lightmid} is $P \otimes (\cdot, \cdot)$ -free, and therefore the number of light-middle 1s in A_{light} is, by superadditivity, at most

$$(3.6) \quad B \cdot \|A_{\text{lightmid}}\|_1 \leq B \cdot \text{Ex}(P \otimes (\cdot, \cdot), n^*/G, m/B) \leq \text{Ex}(P \otimes (\cdot, \cdot), Bn^*/G, m).$$

Consider an occurrence of $(\cdot, \cdot)$ in A_{lightmid} , say $A_{\text{lightmid}}(\ell_j, i) = A_{\text{lightmid}}(\ell_{j'}, i) = 1$. By construction they lie in different chunks, thus there must be a column in slab i of A_{light} that contains two light-middle 1s in blocks $(\ell_j, i), \dots, (\ell_{j'}, i)$ inclusive. Together with a light-first and light-last 1, this forms a $(\cdot, \cdot)$ pattern. Thus, any occurrence of $P \otimes (\cdot, \cdot)$ in A_{lightmid} implies an occurrence of $Q = P \otimes (\cdot, \cdot)$ in A_{light} , contradicting the fact that A_{light} is $Q_{a,b}$ -free.

Combining Eqns. (3.2)(3.3)(3.4)(3.5)(3.6), we arrive at a recursive upper bound on $\text{Ex}(Q_{a,b}, n, m)$.

$$\begin{aligned}
 \text{Ex}(Q_{a,b}, n, m) &\leq \sum_{i=1}^{\lfloor m/B \rfloor} \text{Ex}(Q_{a,b}, n_i, m_i) && \text{local 1s} \\
 &+ \text{Ex}(Q_{a,b+1}, n^*, m - m_{\lfloor m/B \rfloor}) + \text{Ex}(Q_{a+1,b}, n^*, m - m_1) && \text{first and last 1s} \\
 &+ \text{Ex}(P, n^*/G, m/B) \cdot \text{Ex}(Q_{a+1,b+1}, G, B) && \text{heavy middle 1s} \\
 &+ 2 \text{Ex}(Q_{a,b}, n^*, m/B) && \text{light-first/-last 1s} \\
 (3.7) \quad &+ \text{Ex}(P \otimes (\bullet), Bn^*/G, m). && \text{light-middle 1s}
 \end{aligned}$$

3.2 Analysis of The Recurrence

LEMMA 3.1. Let $t = 3k - (a + b)$ be the number of 1s in $Q_{a,b}$. If $t = 3$ then $\text{Ex}(Q_{a,b}, n, m) \leq 2n + (2k - 1)(m - 2)$ and if $t = 2$ then $\text{Ex}(Q_{a,b}, n, m) \leq n + (2k - 1)(m - 1)$.

Proof. First consider $t = 3$. $Q_{a,b}$ contains only three 1s and either $2k - 2$ or $2k - 3$ all-zero rows. Those three 1s are equivalent to $(\cdot, \cdot)$, $(\cdot, \cdot)$, or $(\cdot, \cdot)$. Suppose A is $Q_{a,b}$ -free. Remove the first and last 1 in each row of A , then remove the first $2k - 1$ 1s in each of $m - 2$ columns, excluding the first and last, which are now all zero. If any 1 remains, then there must have been an occurrence of $Q_{a,b}$ in A . The $t = 2$ case is proved similarly. $\square$

LEMMA 3.2. If $m \leq 2^j$, $\text{Ex}(Q_{a,b}, n, m) \leq 2^{t-2}n + (2k - 1)j^{\max\{0, t-3\}}(m - 2)$, where $t = 3k - (a + b)$.

Proof. The cases $t \in \{2, 3\}$ follow from Lemma 3.1 so we may assume $t > 3$. We consider a simplified version of (3.7) in which $B = \lfloor m/2 \rfloor$, i.e., $m_1 = \lfloor m/2 \rfloor$ and $m_2 = \lfloor m/2 \rfloor$. There are only two slabs, all 1s are classified as *local*, *first*, or *last*, and we have

$$\text{Ex}(Q_{a,b}, n, m) \leq \sum_{i \in \{1, 2\}} \text{Ex}(Q_{a,b}, n_i, m_i) + \text{Ex}(Q_{a,b+1}, n^*, m_1) + \text{Ex}(Q_{a+1,b}, n^*, m_2).$$

Applying the inductive hypothesis to each term, this is at most

$$\begin{aligned}
 &\leq 2^{t-2}(n_1 + n_2) + (2k - 1)(j - 1)^{t-3}(\lfloor m/2 \rfloor - 2 + \lfloor m/2 \rfloor - 2) \\
 &\quad + 2 \cdot 2^{t-3}n^* + (2k - 1)(j - 1)^{t-4}(\lfloor m/2 \rfloor - 2 + \lfloor m/2 \rfloor - 2) \\
 &= 2^{t-2}n + (2k - 1)((j - 1)^{t-3} + (j - 1)^{t-4})(m - 4) \\
 &\leq 2^{t-2}n + (2k - 1)j^{t-3}(m - 2).
 \end{aligned}$$

$\square$

We use the following version of Ackermann's function and its inverses.

$$\begin{aligned}
 a_{1,j} &= 2^j && \text{for } j \geq 1, \\
 a_{i,1} &= 2 && \text{for } i \geq 2, \\
 a_{i,j} &= w \cdot a_{i-1,w}, \text{ where } w = a_{i,j-1}. && \text{for } i, j \geq 2,
 \end{aligned}$$

$$\begin{aligned}
 \alpha(n, m) &= \min\{i : a_{i,j} \geq m, \text{ where } j = \max\{3, \lfloor n/m \rfloor\}\} \\
 \alpha(n) &= \alpha(n, n)
 \end{aligned}$$

Observe that in the table of Ackermann values, the 1st column is constant ($a_{i,1} = 2$) and the second merely exponential ($a_{i,2} = 2^{i+1}$) so we have to look to the third column to see Ackermann-type growth, which is why we set j as $j = \max\{3, \lfloor n/m \rfloor\}$.

LEMMA 3.3. Fix a constant $c = 3k$. Suppose $m \leq (a_{i,j})^c$. Then

$$\text{Ex}(Q_{a,b}, n, m) \leq \mu_{i,t}(n + (cj)^{\max\{0, t-3\}}(2k - 1)(m - 2)),$$

where $t = 3k - (a + b)$ and $\mu_{i,t} = (2^{O(kt)} + O(i)^{t-2})2^i$.

Proof. The proof is by induction on i, j , and t . The cases $t \in \{2, 3\}$ were already handled, so assume $t \geq 4$. Let A be a $Q_{a,b}$ -free $n \times m$ matrix, where $m \leq (a_{i,j})^c$. We apply Eqn. (3.7) with B, G set as follows:

$$B = a_{i,j-1}^c,$$

$$G = (c(j-1))^{\max\{0, t-5\}}(2k-1)(B-2).$$

Observe that

$$m/B \leq (a_{i,j}/a_{i,j-1})^c = (a_{i-1, a_{i,j-1}})^c.$$

We apply the induction hypothesis at $(i, j-1, t)$ to local 1s at $(i, j, t-1)$ to first/last 1s, at $(i, j-1, t-2)$ to heavy middle 1s, and at $(i-1, a_{i,j-1}, t)$ to light-first/light-last 1s. Plugging these bounds into Eqn. (3.7) and applying Theorem 2.1 yields the following upper bound.

$$\begin{aligned} \text{Ex}(Q_{a,b}, n, m) & \leq \mu_{i,t}(n - n^*) + \mu_{i,t}(c(j-1))^{t-3}(2k-1)(m - 2m/B) && \text{local} \\ & + 2\mu_{i,t-1}n^* + 2\mu_{i,t-1}(cj)^{t-4}(2k-1)(m-2) && \text{first/last} \\ & + C_k(n^*/G + m/B) \left(\mu_{i,t-2}G + \mu_{i,t-2}(c(j-1))^{\max\{0, t-5\}}(2k-1)(B-2) \right) && \text{heavy} \\ & + 2\mu_{i-1,t}n^* + 2\mu_{i-1,t}(c(a_{i,j-1}))^{t-3}(2k-1)(m/B-2) && \text{light-first/last} \\ & + C'_k(Bn^*/G + m) && \text{light-middle} \end{aligned}$$

Note that by choice of G , the line for heavy 1s is exactly $2C_k\mu_{i,t-2}(n^* + Gm/B)$. Continuing,

$$(3.8) \quad \leq \mu_{i,t}(n + (cj)^{t-3}(2k-1)(m-2))$$

$$(3.9) \quad + [-\mu_{i,t} + 2\mu_{i,t-1} + 2C_k\mu_{i,t-2} + 2\mu_{i-1,t} + C'_k]n^*$$

$$(3.10) \quad + \left[-\mu_{i,t}c(cj)^{t-4} + 2\mu_{i,t-1}(cj)^{t-4} + 2C_k\mu_{i,t-2}(c(j-1))^{\max\{0, t-5\}} \right. \\ \left. + 2\mu_{i-1,t}c^{t-3}a_{i,j-1}^{(t-3)-c} + C'_k \right](2k-1)(m-2)$$

$$(3.11) \quad \leq \mu_{i,t}(n + (cj)^{t-3}(2k-1)(m-2)).$$

Lines (3.8)-(3.10) follow from the fact that $(c(j-1))^{t-3} \leq (cj)^{t-3} - c(cj)^{t-4}$. Line (3.11) completes the induction so long as the bracketed terms in Lines (3.9)-(3.10) are non-positive. These will hold whenever Eqns. (3.12)-(3.13) hold.

$$(3.12) \quad \mu_{i,t} \geq 2\mu_{i,t-1} + 2C_k\mu_{i,t-2} + 2\mu_{i-1,t} + C'_k,$$

$$(3.13) \quad \mu_{i,t} \geq \frac{2\mu_{i,t-1}}{c} + \frac{2C_k\mu_{i,t-2}}{c} + \frac{2\mu_{i-1,t}}{2^{3k-1}} + \frac{C'_k}{2^{t-4}c^{t-3}}.$$

Eqn. (3.13) was obtained by dividing through by $c(cj)^{t-4}$ and noting that $j \geq 2$ and $a_{i,j-1} \geq 2$. Clearly any values $(\mu_{i,t})_{i \geq 1, t \geq 0}$ that satisfy Eqn. (3.12) also satisfy (3.13) so we may focus solely on the former. We argue that the lemma is satisfied for $\mu_{i,t}$ defined as follows. Let $C = C'_k \geq C_k$.

$$(3.14) \quad \mu_{i,t} = (2C + 3i)^{t-2}(2^i - 1).$$

When $t \in \{2, 3\}$ the claim follows from Lemma 3.1 since $\mu_{i,3} \geq 2$ and $\mu_{i,2} \geq 1$. When $i = 1$ and $t \geq 4$, $m \leq (a_{1,j})^c = a_{1,cj} = 2^{cj}$ and the claim follows from Lemma 3.2 since $\mu_{i,t} \geq 2^{t-2}$. Now suppose $i \geq 2, t \geq 4$.

$$\begin{aligned} & 2\mu_{i,t-1} + 2C_k\mu_{i,t-2} + 2\mu_{i-1,t} + C'_k \\ & \leq 2(2C + 3i)^{t-3}(2^i - 1) + 2C(2C + 3i)^{t-4}(2^i - 1) + 2(2C + 3(i-1))^{t-2}(2^{i-1} - 1) + C \\ & \leq (2C + 3i)^{t-2}(2^i - 1) \left(\frac{2}{2C + 3i} + \frac{2C}{(2C + 3i)^2} + 1 - \frac{3}{2C + 3i} \right) \\ & \leq (2C + 3i)^{t-2}(2^i - 1) \left(\frac{2}{2C + 3i} + \frac{1}{2C + 3i} + 1 - \frac{3}{2C + 3i} \right) \\ & \leq (2C + 3i)^{t-2}(2^i - 1) = \mu_{i,t}. \end{aligned}$$

The first inequality is from the inductive hypothesis and $C_k \leq C'_k \leq C$. The second inequality follows from $(2C + 3(i-1))^{t-2} \leq (2C + 3i)^{t-2} - 3(2C + 3i)^{t-3}$. This completes the induction. $\square$

Proof. [Proof of Theorem 1.3] Let A be a Q -free $n \times m$ matrix and $t = c = 3k$. Take i to be minimal such that for $j = \max\{3, \lceil n/m \rceil^{1/t}\}$, $m \leq (a_{i,j})^c$. It is tedious, but straightforward, to show that $i = \alpha(n, m) \pm O(1)$. Lemma 3.3 bounds the number of 1s in A by

$$\begin{aligned} \mu_{i,t}(n + (cj)^{t-3}(2k-1)m) &= \mu_{i,t}\left(n + 2^{O(k \log k)}n\right) & (cj)^{t-3} &< 2^{O(k \log k)}(n/m) \\ &= n \cdot 2^{O(k \log k)} \cdot (2C + 3i)^{t-2} 2^i \\ &= n \cdot \left(2^{O(k^2)} + O(i)^{3k-2}\right) 2^i & C &= 2^{O(k)}; \text{ see Theorem 2.1} \\ &= n \cdot 2^{O(k^2) + (1+o(1))\alpha(n,m)}. \end{aligned}$$

$\square$

4 Lower Bounds on 0–1 Matrices via Sequences

Blocked Sequences and 0–1 Matrices. If S is a sequence, let $|S|$ be its length and $\|S\|$ the size of its alphabet $|\Sigma(S)|$. A *block* is a contiguous sequence of distinct symbols. If S is understood to be partitioned into blocks, $\llbracket S \rrbracket$ is the number of blocks. Regardless of $\Sigma(S)$, we can always write S in *canonical form* over the alphabet $\{1, \dots, \|S\|\}$, where the symbols are sorted according to their first appearance in S . If S is in canonical form, its canonical matrix A_S is the $\|S\| \times \llbracket S \rrbracket$ symbol-block incidence matrix, i.e., $A_S(i, j) = 1$ if symbol i appears in block j , and 0 otherwise. One cannot quite recover S from A_S since A_S does not encode the order of symbols within a block. Nonetheless, the transformation is *useful* inasmuch as subsequences avoided by S often become 0–1 patterns avoided by A_S .

Composition and Shuffling. We consider sequences S partitioned into *live* and *dead* blocks satisfying extra constraints:

- All live blocks have the same length. Dead blocks have variable lengths, and the number of dead blocks between consecutive live blocks is also variable.
- The first occurrence of every symbol appears in a dead block, and dead blocks contain only first occurrences. Let $\llbracket S \rrbracket$ be the number of live blocks in S .

Composition. Suppose U_{top} is a sequence in which all live blocks have length j and U_{mid} is a sequence with $\|U_{\text{mid}}\| = j$. The *composition* $U_{\text{sub}} = U_{\text{top}} \circ U_{\text{mid}}$ is obtained by replacing each live block L of U_{top} with a copy $U_{\text{mid}}(L)$ over the alphabet of L , whereas dead blocks of U_{top} are inherited by U_{sub} verbatim. In general $U_{\text{mid}}(L)$ can contain both live and dead blocks [Pet15b], but in our particular construction $U_{\text{mid}}(L)$ contains only live blocks.

Shuffling. Now suppose U_{sub} is a sequence whose live blocks have length j and U_{bot} is a sequence with $\|U_{\text{bot}}\| = j$. The *left-shuffle* $U_{\text{sub}} \otimes U_{\text{bot}}$ is obtained as follows. Let $U_{\text{sub}} = D_0 L_1 D_1 L_2 D_2 \cdots L_k D_k$, where L_i is the i th live block, D_i is zero or more dead blocks, and $k = \llbracket U_{\text{sub}} \rrbracket$. Let $U_{\text{bot}}^* = U_{\text{bot}}^{(1)} \cdots U_{\text{bot}}^{(k)}$ be the concatenation of k copies of U_{bot} over disjoint alphabets. The *left-shuffle* is obtained by taking, for all i , the block $L_i = (a_1 a_2 \cdots a_j)$ and inserting a_ℓ , for $\ell \in [1, j]$, at the *left end* of the ℓ th live block of $U_{\text{bot}}^{(i)}$, then inserting dead blocks D_i between $U_{\text{bot}}^{(i)}$ and $U_{\text{bot}}^{(i+1)}$.⁴

Sequence Construction. $U(j)$ and $U(i, j)$ are blocked sequences, where square brackets indicate dead blocks and parentheses indicate live blocks. $U(i, j)$ is a variation on order-4 Davenport-Schinzel sequences [ASS89], adapted specifically to exclude a small pattern that arises from $P \otimes (\cdot, \cdot)$ -type patterns.

⁴The *right shuffle* $U_{\text{sub}} \otimes U_{\text{bot}}$ is defined in the same way, except that a_ℓ is inserted at the *right end* of the ℓ th live block of $U_{\text{bot}}^{(i)}$. We only use left-shuffles but there are cases where it is desirable to use both [Pet11b, Pet15b].

$$\begin{aligned}
 U(j) &= (j(j-1) \cdots 1)(12 \cdots j) && \text{2 live blocks} \\
 U(1, j) &= 12 \cdots j && \text{first block dead, second live} \\
 U(i, 0) &= ()^2 && \text{2 empty live blocks} \\
 U(i, j) &= (U_{\text{top}} \circ U_{\text{mid}}) \otimes U_{\text{bot}} \\
 \text{where } U_{\text{top}} &= U(i-1, \llbracket U(i, j-1) \rrbracket), \\
 U_{\text{mid}} &= U(\llbracket U(i, j-1) \rrbracket), \\
 \text{and } U_{\text{bot}} &= U(i, j-1)
 \end{aligned}$$

Let $N(i, j) = \|U(i, j)\|$ be the alphabet size and $L(i, j) = \llbracket U(i, j) \rrbracket$ be the number of live blocks. N, L obey the following recurrence:

$$\begin{aligned}
 L(1, j) &= 1 \\
 L(i, 0) &= 2 \\
 L(i, j) &= L(i, j-1) \cdot 2 \cdot L(i-1, L(i, j-1)) \\
 N(1, j) &= j \\
 N(i, j) &= N(i, j-1) \cdot 2 \cdot L(i-1, L(i, j-1)) + N(i-1, L(i, j-1))
 \end{aligned}$$

LEMMA 4.1. Fix any $U = U(i, j)$.

1. All live blocks in U have length j . Each symbol appears $2^{i-1} + 1$ times in U , its first occurrence appearing in a dead block, and the remaining 2^{i-1} times in live blocks.
2. As a consequence of part 1, $N(i, j) = (j/2^{i-1}) \cdot L(i, j)$.
3. The number of dead blocks is at most $L(i, j) - 1$.
4. If $n = N(i, j)$ is the number of symbols in U and $m < 2L(i, j)$ the number of blocks, $i = \alpha(n, m) \pm O(1)$, and $|U| = \Theta(n2^{\alpha(n, m)})$.

Proof. Part 1. The claim holds in the base cases $U(1, j)$ and $U(i, 0)$. All live blocks in $U_{\text{bot}} = U(i, j-1)$ have length $j-1$ by induction, and each receive one symbol in the shuffling operation $(U_{\text{top}} \circ U_{\text{mid}}) \otimes U_{\text{bot}}$. All symbols in $U_{\text{bot}} = U(i, j-1)$ appear in 2^{i-1} live blocks. Those in $U_{\text{top}} = U(i-1, \llbracket U(i, j-1) \rrbracket)$ appear in 2^{i-2} live blocks, and therefore in 2^{i-1} live blocks in $U_{\text{sub}} = U_{\text{top}} \circ U_{\text{mid}}$ since U_{mid} doubles the number of live occurrences. The property that first occurrences appear in dead blocks is preserved by composition and shuffling. *Part 2.* Note that both $j \cdot L(i, j)$ and $2^{i-1} \cdot N(i, j)$ both count the total length of all live blocks. *Part 3.* The claim holds in all base cases. By induction, the number of dead blocks in U_{top} is at most $L(i-1, L(i, j-1)) - 1$. U_{bot}^* consists of $2L(i-1, L(i, j-1))$ copies of $U_{\text{bot}} = U(i, j-1)$, so U_{bot}^* has $2L(i-1, L(i, j-1))(L(i, j-1) - 1)$ dead blocks. In total there are $L(i-1, L(i, j-1))(2L(i, j-1) - 1) - 1 \leq L(i, j) - 1$ dead blocks. *Part 4.* Proving Ackermann-like functions are equivalent inasmuch as their inverses differ by $\pm O(1)$ is tedious, but straightforward. See [Pet06, Lemma 3.10] for an example of such a proof. $\square$

LEMMA 4.2. Let $U = U(i, j)$ be obtained from $U_{\text{top}}, U_{\text{mid}}, U_{\text{bot}}$. Suppose $a < b$ are two symbols in $\Sigma(U)$ appearing in a common live block.

1. The restriction of U to letters $\{a, b\}$ is of the form $a^*b^*(ab)b^*a^*$.
2. If $a \in \Sigma(U_{\text{top}})$, $b \in \Sigma(U_{\text{bot}}^*)$, then $a < c$ for every symbol c appearing in b 's copy of U_{bot} .

Proof. The claim is true in the base cases $U(1, j)$ and $U(i, 0)$. Consider the moment that a is shuffled into b 's live block, where $a \in \Sigma(U_{\text{top}})$ and $b \in \Sigma(U_{\text{bot}}^*)$. All occurrences of b appear in one copy of U_{bot} in U_{bot}^* , and exactly one occurrence of a is shuffled into this copy. It follows that the restriction of U to letters $\{a, b\}$ is of the form $a^*|b^*(ab)b^*|a^*$, where the bars mark the boundary of b 's copy of U_{bot} . Furthermore, since the first occurrence of a is in a dead block, which is inserted between two copies of U_{bot} in U_{bot}^* , $a < c$ for every c in b 's copy of U_{bot} . $\square$

LEMMA 4.3. $U = U(i, j)$ does not contain any subsequences order-isomorphic to 41213.

Proof. Since U is in canonical form, the existence of 41213 implies the existence of a subsequence order-isomorphic to

$$\sigma = 31213$$

Suppose that σ first appears in $U(i, j) = U_{\text{sub}} \otimes U_{\text{bot}} = (U_{\text{top}} \circ U_{\text{mid}}) \otimes U_{\text{bot}}$. If σ already appears in U_{sub} but did not appear in U_{top} , then U_{top} must have contained σ' .

$$\sigma' = 31(12)3$$

Note that $\{2, 3\}$ cannot share a live block in U_{top} without also including 1, and if $\{1, 2, 3\}$ shared a live block in U_{top} , the restriction of U_{top} to $\{1, 2, 3\}$ would, by Lemma 4.2(1), be:

$$1^*2^*3^*(123)3^*2^*1^*$$

and the restriction of U_{sub} to $\{1, 2, 3\}$ would be:

$$1^*2^*3^*(321)(123)3^*2^*1^*,$$

which does not contain σ . We therefore need to argue that neither σ nor σ' can arise in $U(i, j)$ in the *shuffling* operation.

If σ or σ' arose during shuffling, then Lemma 4.2 implies that for any $a, b \in \{1, 2, 3\}$ with $a \in \Sigma(U_{\text{top}})$ and $b \in \Sigma(U_{\text{bot}}^*)$, that $a < b$. It cannot be that $\{1\}$ or $\{1, 2\} \subset \Sigma(U_{\text{top}})$ while $\{2, 3\}$ or $\{3\} \subset \Sigma(U_{\text{bot}}^*)$ since 3's copy of U_{bot} only receives *one* copy of any symbol during shuffling, but both σ, σ' have two 1s between the first and last 3. $\square$

REMARK 4.1. The distinction between live and dead blocks is critical for constructing order-3 (ababa-free) Davenport-Schinzel sequences [HS86, Niv10, Pet15a], and generalized DS sequences with length $O(n \text{ poly}(\alpha(n)))$ [Pet11b, Pet15b]. However, all constructions of DS sequences at order-4 and above [ASS89, Niv10, Pet15a] (having length $\Omega(n2^{\alpha(n)})$) only use live blocks. In Lemma 4.3, it is very important that first occurrences lie exclusively in dead blocks, and are never shuffled into the middle of a copy of U_{bot} . If the first block in $U(1, j)$ were redefined to be live, then we would see instances of 41213 in $U(i, j)$. It could be that 12121 appears in a copy of U_{bot} , and the first occurrences of $\{3, 4\}$ lie in a common live block in U_{top} . The restriction of U_{top} to $\{3, 4\}$ contains (34)3. After shuffling the block (34) into the U_{bot} containing 1, 2 we can see 12341213. Lemma 4.2(2) rules out this possibility when first occurrences appear in dead blocks.

THEOREM 4.1. $\text{Ex}(W, n, m) = \Theta(m + n2^{\alpha(n, m)})$, where

$$W = \begin{pmatrix} \cdot & & \cdot \\ & \cdot & \\ & & \cdot \end{pmatrix}$$

Proof. By Lemma 4.1, the sequence $U = U(i, j)$ has $n = N(i, j)$ symbols, $m < 2L(i, j)$ blocks, and length $|U| = \Theta(n2^i) = \Theta(n2^{\alpha(n, m)})$. We convert U to an $n \times m$ 0–1 matrix A_U . Number the rows of A_U from bottom-to-top, and the columns from left-to-right, and let $A_U(i, j) = 1$ iff symbol i appears in block j . U does not contain subsequences order-isomorphic to 41213, which implies that A_U is W -free, and hence $\text{Ex}(W, n, m) = \Omega(n2^{\alpha(n, m)})$. The matching upper bound is obtained as in [Pet11b, Thm. 3.4]. $\square$

5 Additional Upper Bounds

5.1 Proof of Theorem 1.4 Recall that I_k is the $k \times k$ identity matrix. For example, when $k = 3$, we have the following.

$$I_k = \begin{pmatrix} \cdot & & \\ & \cdot & \\ & & \cdot \end{pmatrix} \quad I_k \otimes (\cdot, \cdot) = \begin{pmatrix} \cdot & \cdot & & \\ \cdot & \cdot & & \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & & & \cdot & \cdot \end{pmatrix}$$

Theorem 1.4 follows from Keszegh's [Kes09] *joining* operation and Pettie's upper bound on order-3 Davenport-Schinzel sequences [Pet15a]; cf. [HS86, Niv10]. Keszegh [Kes09] proved that if R has a 1 in its southeast corner and S has a 1 in its northwest corner, that $\text{Ex}(R \oplus S, n, m) \leq \text{Ex}(R, n, m) + \text{Ex}(S, n, m)$, where $R \oplus S$ is formed by joining R, S at their corners.

$$R \oplus S = \left(\begin{array}{c} \boxed{\begin{array}{c} R \end{array}} \quad \begin{array}{c} \bullet \\ \boxed{\begin{array}{c} S \end{array}} \end{array} \right).$$

Observe that $I_k \otimes (\cdot, \cdot) = (I_{k-1} \otimes (\cdot, \cdot)) \oplus (\cdot, \cdot)$, so we can apply Keszegh's operation $k - 2$ times to reduce to the base case $I_2 \otimes (\cdot, \cdot)$. We claim $\text{Ex}(I_2 \otimes (\cdot, \cdot), n, m) < \text{Ex}((\cdot, \cdot), n, m) + 2n + m$. Suppose A is $I_2 \otimes (\cdot, \cdot)$ -free, and let A' be obtained by removing the top 1 in each column, and then the first two 1s in each row. Then A' is clearly $(\cdot, \cdot)$ -free. Putting it all together, we have,

$$\begin{aligned} \text{Ex}(I_k \otimes (\cdot, \cdot), n, m) &\leq \text{Ex}(I_2 \otimes (\cdot, \cdot), n, m) + (k - 2) \text{Ex}((\cdot, \cdot), n, m) \\ &\leq (k - 1) \text{Ex}((\cdot, \cdot), n, m) + 2n + m \\ &\leq (k - 1)2n\alpha(n, m) + O(k(n + m)), \end{aligned}$$

where the last inequality follows from the bound $\text{Ex}((\cdot, \cdot), n, m) = 2n\alpha(n, m) + O(n + m)$ on order-3 Davenport-Schinzel sequences [Pet15a].

5.2 Avoiding W and Its Reflection By symmetry, Theorem 4.1 also applies to $\text{Ex}(W', n, m)$, where W' is the reflection of W along the y -axis. However, the density of $\{W, W'\}$ -free matrices is asymptotically smaller.

THEOREM 5.1. $\text{Ex}(\{W, W'\}, n, m) \leq 4n\alpha(n, m) + O(n + m)$, where W' is the reflection of W along the y -axis.

$$W' = \begin{pmatrix} & & & \bullet \\ \bullet & & & \\ & \bullet & & \\ & & \bullet & \end{pmatrix}$$

Proof. Let A be a $\{W, W'\}$ -free matrix. Remove the top 1 in each column, yielding A' . It follows that A' is $\{W, W', W''\}$ -free, where

$$W'' = \begin{pmatrix} \bullet & & & \bullet \\ & \bullet & & \\ & & \bullet & \\ & & & \bullet \end{pmatrix}$$

We prove that $\text{Ex}(\{W, W', W''\}, n, m) \leq 2\text{Ex}((\cdot, \cdot), n, m)$. Call a 1 in A' *bottom-right* if it appears as the bottom-right 1 in a copy of $(\cdot, \cdot)$, and *bottom-left* if it appears as the bottom-left 1 in a copy of $(\cdot, \cdot)$. If $\|A'\|_1 \geq 2\text{Ex}((\cdot, \cdot), n, m) + 1$ then some $A'(i, j) = 1$ must be classified as both bottom-left and bottom-right. Let (i_L, j_L) and (i_R, j_R) be the positions of the top-left 1 in a copy of $(\cdot, \cdot)$ containing $A'(i, j)$ and top-right 1 in a copy of $(\cdot, \cdot)$ containing $A'(i, j)$, respectively. Numbering the rows from bottom to top, we have a copy of W'' if $i_L = i_R$, a copy of W' if $i_L < i_R$, and a copy of W if $i_L > i_R$. For example, when $i_L < i_R$,

$$W'' = \begin{pmatrix} & & & \bullet & \\ & & & & (i_R, j_R) \\ & \bullet & & & \\ (i_L, j_L) & & & & \\ & \bullet & & & \\ & & \bullet & & \\ & & & \bullet & \\ & & & & (i, j) \end{pmatrix}$$

It is known [Pet15a, Niv10] that $\text{Ex}((\cdot, \cdot), n, m) = 2n\alpha(n, m) \pm O(n + m)$. $\square$

6 Concluding Remarks

Derryberry et al. [DSW05], Demaine et al. [DHI⁺09], and Pettie [Pet10] developed the idea of representing the behavior of a data structure as a point-set in $[m] \times [n]$, or equivalently, a 0–1 matrix in $\{0, 1\}^{m \times n}$, in which the axes correspond to *time* and *space*. Applying results on forbidden 0–1 matrices in this context is very natural, and has led to sharp or nearly sharp bounds on certain path compression schemes [Pet10], structured inputs to binary search trees [CGK⁺15b, CGK⁺15a, CGJ⁺23, Pet10], and heaps [KS20].

Although the forbidden 0–1 matrix framework is perfectly suited to proving that sorting π -free sequences takes *near-linear* time, it might be inadequate to establishing an optimal $O_k(n)$ time bound. As we have shown, any $o(n2^{\alpha(n)})$ -time analysis must use some property *beyond* $P_\pi \otimes (\cdot, \cdot)$ -freeness. Here it may be useful to consider different ways to decompose a 0–1 matrix; see Guillemot and Marx [GM14] and Chalermsook, Gupta, Jiamjitrak, Acosta, Pareek, and Yingchareonthawornchai [CGJ⁺23].

The literature on forbidden 0–1 matrices is rich [FH92, Tar05, PT06, Pet11b, Pet11c, Pet11a, Pet15b, Kes09, Gen09, Ful09, KTTW19, CK12, CK17, Fox13, MT04, Kla92, PT24] but there are many outstanding open problems. In the context of *data structure analysis*, the most interesting open problems are to characterize the set of *linear* forbidden patterns—those P with $\text{Ex}(P, n) = O(n)$ —and in particular, to characterize linear *light* patterns. It is known that there are infinitely many minimal (with respect to $<$) non-linear patterns [Kes09, Gen09, Pet11a], but there may be other ways to characterize this set in a finite representation. On the other hand, we know of only two minimally non-linear *light* patterns (with respect to $<$ and reflections), namely $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$ and $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$. It is quite possible that these are the only sources of non-linearity in light patterns.

References

- [AB15] Michael Albert and Mireille Bousquet-Mélou. Permutations sortable by two stacks in parallel and quarter plane walks. *Eur. J. Comb.*, 43:131–164, 2015.
- [AMR02] Mike D. Atkinson, Max Murphy, and Nikola Ruskuc. Sorting with two ordered stacks in series. *Theor. Comput. Sci.*, 289(1):205–223, 2002.
- [Art07] David Arthur. Fast sorting and pattern-avoiding permutations. In *Proceedings of the Fourth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 169–174, 2007.
- [ASS89] Pankaj K. Agarwal, Micha Sharir, and Peter W. Shor. Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences. *J. Combin. Theory Ser. A*, 52:228–274, 1989.
- [BGH⁺10] Therese Biedl, Alexander Golynski, Angèle M. Hamel, Alejandro López-Ortiz, and J. Ian Munro. Sorting with networks of data structures. *Discret. Appl. Math.*, 158(15):1579–1586, 2010.
- [BKO23] Benjamin Aram Berendsohn, László Kozma, and Michal Opler. Optimization with pattern-avoiding input. *CoRR*, abs/2310.04236, 2023.
- [Bón02] Miklós Bóna. A survey of stack-sorting disciplines. *Electron. J. Comb.*, 9(2), 2002.
- [Bón22] Miklós Bóna. *Combinatorics of Permutations, 3rd Ed.* CRC Press, New York, 2022.
- [CGJ⁺23] Parinya Chalermsook, Manoj Gupta, Wanchote Jiamjitrak, Nidia Obscura Acosta, Akash Pareek, and Sorrachai Yingchareonthawornchai. Improved pattern-avoidance bounds for greedy BSTs via matrix decomposition. In *Proceedings of the 34th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 509–534, 2023.
- [CGK⁺15a] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Greedy is an almost optimal deque. In *Proceedings 14th International Symposium on Algorithms and Data Structures (WADS)*, volume 9214 of *Lecture Notes in Computer Science*, pages 152–165, 2015.
- [CGK⁺15b] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Pattern-avoiding access in binary search trees. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 410–423, 2015.
- [Cib09] Josef Cibulka. On constants in the Füredi-Hajnal and the Stanley-Wilf conjecture. *J. Comb. Theory, Ser. A*, 116(2):290–302, 2009.
- [CK12] Josef Cibulka and Jan Kynčl. Tight bounds on the maximum size of a set of permutations with bounded VC-dimension. *J. Combin. Theory Ser. A*, 119(7):1461–1478, 2012.
- [CK17] Josef Cibulka and Jan Kynčl. Better upper bounds on the Füredi-Hajnal limits of permutations. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2280–2293, 2017.
- [CMSS00] Richard Cole, Bud Mishra, Jeanette P. Schmidt, and Alan Siegel. On the dynamic finger conjecture for splay trees I: Splay sorting $\log n$ -block sequences. *SIAM J. Comput.*, 30(1):1–43, 2000.

- [Col00] Richard Cole. On the dynamic finger conjecture for splay trees II: The proof. *SIAM J. Comput.*, 30(1):44–85, 2000.
- [DHI⁺09] Erik D. Demaine, Dion Harmon, John Iacono, Daniel M. Kane, and Mihai Pătraşcu. The geometry of binary search trees. In *Proc. 20th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 496–505, 2009.
- [DSW05] J. Derryberry, D. D. Sleator, and C. C. Wang. A lower bound framework for binary search trees with rotations. Technical Report CMU-CS-05-187, Carnegie Mellon University, 2005.
- [EG17a] Andrew Elvey Price and Anthony J. Guttmann. Permutations sortable by deque and by two stacks in parallel. *Eur. J. Comb.*, 59:71–95, 2017.
- [EG17b] Andrew Elvey Price and Anthony J. Guttmann. Permutations sortable by two stacks in series. *Adv. Appl. Math.*, 83:81–96, 2017.
- [FH92] Zoltán Füredi and Péter Hajnal. Davenport-Schinzel theory of matrices. *Discrete Mathematics*, 103(3):233–251, 1992.
- [Fox13] Jacob Fox. Stanley-Wilf limits are typically exponential. *CoRR*, abs/1310.8378, 2013.
- [FP08] Stefan Felsner and Martin Pergel. The complexity of sorting with networks of stacks and queues. In *Proceedings of the 16th Annual European Symposium on Algorithms (ESA)*, volume 5193 of *Lecture Notes in Computer Science*, pages 417–429, 2008.
- [Fre76] Michael L. Fredman. How good is the information theory bound in sorting? *Theoretical Computer Science*, 1(4):355–361, 1976.
- [FSST86] Michael L. Fredman, Robert Sedgewick, Daniel D. Sleator, and Robert E. Tarjan. The pairing heap: a new form of self-adjusting heap. *Algorithmica*, 1(1):111–129, 1986.
- [Ful09] Radoslav Fulek. Linear bound on extremal functions of some forbidden patterns in 0–1 matrices. *Discrete Mathematics*, 309:1736–1739, 2009.
- [Gen09] Jesse T. Geneson. Extremal functions of forbidden double permutation matrices. *J. Combin. Theory Ser. A*, 116(7):1235–1244, 2009.
- [Gen15] Jesse T. Geneson. *Bounds on extremal functions of forbidden patterns*. PhD thesis, Massachusetts Institute of Technology, Department of Mathematics, 2015.
- [GG19] Navin Goyal and Manoj Gupta. Better analysis of greedy binary search tree on decomposable sequences. *Theor. Comput. Sci.*, 776:19–42, 2019.
- [GM14] Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In *Proceedings 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–101, 2014.
- [GT17] Jesse T. Geneson and Peter M. Tian. Extremal functions of forbidden multidimensional matrices. *Discret. Math.*, 340(12):2769–2781, 2017.
- [HI01] Tero Harju and Lucian Ilie. Forbidden subsequences and permutations sortable on two parallel stacks. In *Where Mathematics, Computer Science, Linguistics and Biology Meet: Essays in Honour of Gheorghe Paun*, pages 267–275. Kluwer Academic Publishers, 2001.
- [HS86] Sergiu Hart and Micha Sharir. Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes. *Combinatorica*, 6(2):151–177, 1986.
- [IL16] John Iacono and Stefan Langerman. Weighted dynamic finger in binary search trees. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 672–691, 2016.
- [Kes09] Balázs Keszegh. On linear forbidden submatrices. *J. Combin. Theory Ser. A*, 116(1):232–241, 2009.
- [Kla92] Martin Klazar. A general upper bound in extremal theory of sequences. *Comment. Math. Univ. Carolin.*, 33(4):737–746, 1992.
- [Kla00] Martin Klazar. The Füredi-Hajnal conjecture implies the Stanley-Wilf conjecture. In *Formal power series and algebraic combinatorics (Moscow, 2000)*, pages 250–255. Springer, Berlin, 2000.
- [Knu73] Donald E. Knuth. *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 1973.
- [KS20] László Kozma and Thatchaphol Saranurak. Smooth heaps and a dual view of self-adjusting data structures. *SIAM J. Comput.*, 49(5), 2020.
- [KTTW19] Dániel Korándi, Gábor Tardos, István Tomon, and Craig Weidert. On the Turán number of ordered forests. *J. Comb. Theory, Ser. A*, 165:32–43, 2019.
- [LT19] Caleb C. Levy and Robert E. Tarjan. Splaying preorders and postorders. In *Proceedings 16th International Symposium on Algorithms and Data Structures (WADS)*, volume 11646 of *Lecture Notes in Computer Science*, pages 510–522. Springer, 2019.
- [Luc88] Joan M. Lucas. Canonical forms for competitive binary search tree algorithms. Technical Report DCS-TR-250, Rutgers University, 1988.
- [Luc91] Joan M. Lucas. On the competitiveness of splay trees: Relations to the union-find problem. In Lyle A. McGeoch and Daniel D. Sleator, editors, *On-line Algorithms*, volume 7 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 95–124, 1991.
- [MSS19] Toufik Mansour, Howard Skogman, and Rebecca Smith. Passing through a stack k times with reversals. *Eur. J.*

- Comb.*, 81:309–327, 2019.
- [MT04] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Combin. Theory Ser. A*, 107(1):153–160, 2004.
- [Mun00] J. Ian Munro. On the competitiveness of linear search. In *Proceedings 8th Annual European Symposium on Algorithms (ESA)*, pages 338–345, 2000.
- [Niv10] Gabriel Nivasch. Improved bounds and new techniques for Davenport-Schinzel sequences and their generalizations. *J. ACM*, 57(3), 2010.
- [Pet06] Seth Pettie. An inverse-Ackermann type lower bound for online minimum spanning tree verification. *Combinatorica*, 26(2):207–230, 2006.
- [Pet08] Seth Pettie. Splay trees, Davenport-Schinzel sequences, and the deque conjecture. In *Proceedings 19th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1115–1124, 2008.
- [Pet10] Seth Pettie. Applications of forbidden 0–1 matrices to search tree- and path compression-based data structures. In *Proceedings 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1457–1467, 2010.
- [Pet11a] Seth Pettie. Degrees of nonlinearity in forbidden 0–1 matrix problems. *Discrete Mathematics*, 311:2396–2410, 2011.
- [Pet11b] Seth Pettie. Generalized Davenport-Schinzel sequences and their 0–1 matrix counterparts. *J. Comb. Theory Ser. A*, 118(6):1863–1895, 2011.
- [Pet11c] Seth Pettie. On the structure and composition of forbidden sequences, with geometric applications. In *Proceedings 27th Annual Symposium on Computational Geometry (SoCG)*, pages 370–379, 2011.
- [Pet15a] Seth Pettie. Sharp bounds on Davenport-Schinzel sequences of every order. *J. ACM*, 62(5):36, 2015.
- [Pet15b] Seth Pettie. Three generalizations of Davenport-Schinzel sequences. *SIAM J. Discrete Mathematics*, 29(4):2189–2238, 2015.
- [PT06] János Pach and Gábor Tardos. Forbidden paths and cycles in ordered graphs and matrices. *Israel J. Math.*, 155:359–380, 2006.
- [PT24] Seth Pettie and Gábor Tardos. On the extremal functions of acyclic forbidden 0–1 matrices. In *Proceedings 35th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2024.
- [ST85] Daniel D. Sleator and Robert E. Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, 1985.
- [Sun92] Rajamani Sundar. On the deque conjecture for the splay algorithm. *Combinatorica*, 12(1):95–124, 1992.
- [Tar72] Robert E. Tarjan. Sorting using networks of queues and stacks. *J. ACM*, 19(2):341–346, 1972.
- [Tar85] Robert E. Tarjan. Sequential access in splay trees takes linear time. *Combinatorica*, 5(4):367–378, 1985.
- [Tar05] Gábor Tardos. On 0–1 matrices and small excluded submatrices. *J. Combin. Theory Ser. A*, 111(2):266–288, 2005.