

# Improved Pattern-Avoidance Bounds for Greedy BSTs via Matrix Decomposition

Parinya Chalermsook<sup>1</sup>, Manoj Gupta<sup>2</sup>, Wanchote Jiamjitrak<sup>1</sup>, Nidia Obscura Acosta<sup>1</sup>,  
Akash Pareek<sup>2</sup> and Sorrachai Yingchareonthawornchai<sup>1</sup>

<sup>1</sup>Aalto University

<sup>2</sup>IIT Gandhinagar

## Abstract

Greedy BST (or simply Greedy) is an online self-adjusting binary search tree defined in the geometric view ([Lucas, 1988; Munro, 2000; Demaine, Harmon, Iacono, Kane, Patrascu, SODA 2009]). Along with Splay trees (Sleator, Tarjan 1985), Greedy is considered the most promising candidate for being dynamically optimal, i.e., starting with any initial tree, their access costs on any sequence is conjectured to be within  $O(1)$  factor of the offline optimal. However, despite having received a lot of attention in the past four decades, the question has remained elusive even for highly restricted input.

In this paper, we prove new bounds on the cost of Greedy in the “pattern avoidance” regime. Our new results include:

- The (preorder) traversal conjecture for Greedy holds up to a factor of  $O(2^{\alpha(n)})$ , improving upon the bound of  $2^{\alpha(n)^{O(1)}}$  in (Chalermsook et al., FOCS 2015)<sup>1</sup>. This is the best known bound obtained by any online BSTs.
- We settle the postorder traversal conjecture for Greedy. Previously this was shown for Splay trees only in certain special cases (Levy and Tarjan, WADS 2019).
- The deque conjecture for Greedy holds up to a factor of  $O(\alpha(n))$ , improving upon the bound  $2^{O(\alpha(n))}$  in (Chalermsook, et al., WADS 2015). This is arguably “one step away” from the bound  $O(\alpha^*(n))$  for Splay trees (Pettie, SODA 2010).
- The split conjecture holds for Greedy up to a factor of  $O(2^{\alpha(n)})$ . Previously the factor of  $O(\alpha(n))$  was shown for Splay trees only in a special case (Lucas, 1988).

The input sequences in traversal and deque conjectures are perhaps “easiest” in the pattern-avoiding input classes and yet among the most notorious special cases of the dynamic optimality conjecture. Key to all these results is to partition (based on the input structures) the execution log of Greedy into several simpler-to-analyze subsets for which classical forbidden submatrix bounds can be leveraged. We believe that this simple method will find further applications in doing amortized analysis of data structures via extremal combinatorics. Finally, we show the applicability of this technique to handle a class of increasingly complex pattern-avoiding input sequences, called *k-increasing sequences*.

As a bonus, we discover a new class of permutation matrices whose extremal bounds are polynomially bounded. This gives a partial progress on an open question by Jacob Fox (2013).

---

<sup>1</sup> $\alpha(n)$  is the inverse Ackermann function of  $n$ .

# 1 Introduction

The dynamic optimality conjecture postulates that there exists an online binary search tree whose cost to serve any input sequence (search, insert, delete) is at most the optimal offline cost of any binary search tree. The two most promising candidates for being dynamically optimal are Splay trees [ST85] and Greedy [DHI<sup>+</sup>09, Mun00, Luc88]. Despite continuing efforts for many decades (see, e.g., the surveys and monographs [Iac13, Koz16, CGK<sup>+</sup>16]), the conjecture remains wide open even for highly restricted corollaries of the conjecture.

This paper focuses on Greedy’s bounds for such corollaries through the lens of “pattern avoidance” (to be made precise later). Each of them is of independent interest and therefore has received a lot of attention in the literature. We describe below some of the most important conjectures that fall under the regime of pattern avoidance:

**(Preorder) Traversal Conjecture:** There exists an online binary search tree that, starting with any initial BST on  $n$  keys, for any input  $X = (x_1, \dots, x_n) \in [n]^n$  obtained by *preorder* traversal of (potentially distinct) binary search tree  $R$  on  $[n]$ , searches  $X$  with  $O(n)$  cost.<sup>a</sup>

<sup>a</sup>More formally,  $x_t$  is searched at time  $t$  for all  $t \in [n]$ .

**(Postorder) Traversal Conjecture:** There exists an online binary search trees that, starting with any initial BST on  $n$  keys, for any input  $X = (x_1, \dots, x_n) \in [n]^n$  obtained by *postorder* traversal of (potentially distinct) binary search tree  $R$  on  $[n]$ , searches  $X$  with  $O(n)$  cost.

**Deque Conjecture:** For  $m > n$ , there exists an online binary search tree that, starting with any initial BST on  $n$  keys, can serve  $m$  operations INSERTMIN, INSERTMAX, DELETEMIN, DELETEMAX in  $O(m)$  time.

**Split Conjecture:** There exists an online binary search tree that, starting with any initial tree with  $n$  keys, serves a sequence of SPLIT operations in time  $O(n)$  where the operation SPLIT( $i$ ) moves  $i$  to the root and then deletes it, leaving two independent binary search (split) trees.

Resolving these conjectures (especially the deque and preorder conjecture) is considered the “simplest” step of dynamic optimality conjecture and yet has so far resisted attempts for past three decades.

One can ask popular conjectures of BST in two settings: (1) when the initial BST can be pre-processed or (2) when it cannot be pre-processed. There is a gap in our understanding of these two settings. For example, it is not known if Greedy’s cost for preorder traversal is same in both the settings. In setting (1), Chalermsook et al. [CGK<sup>+</sup>15b] showed that Greedy takes  $O(n)$  for the preorder traversal *if preprocessing is allowed*. One can also solve this problem using the ideas in [IL16]. For Splay tree, if preprocessing is allowed, then Splay tree takes  $O(n)$  for the preorder traversal [CH93]. We will consider setting (2) in this paper.

**Bounds on Splay Trees.** For Deque Conjecture, Splay has been shown to cost at most  $O(m\alpha(n))$  by Sundar [Sun92] and later  $O(m\alpha^*(n))$  by Pettie [Pet08]. It has remained open whether Splay’s cost is  $o(n \log n)$  for preorder and postorder traversals. Special cases when we start inserting preorder or postorder sequence  $X$  from an empty-initial tree were resolved recently by Levy and Tarjan [LT19]. Lucas [Luc92] showed that the split costs  $O(n\alpha(n))$  in Splay trees when the initial tree is a path.

**Bounds on Greedy.** The bounds known for Greedy are generally better than the Splay’s counterpart (except for Deque). For Deque Conjecture, Greedy is known to cost  $m2^{O(\alpha(m, m+n))}$  [CGK<sup>+</sup>15a]. For both preorder and postorder traversal sequences, Greedy is known to cost at most  $n2^{\alpha(n)^{O(1)}}$

[CGK<sup>+</sup>15b] (where the constant in the exponent is relatively large). We are not aware of published results for Split conjecture.

**Broader context: Amortized analysis and forbidden submatrix theory.** Resolving these conjectures represents a small part of a much broader algorithmic challenge in amortized analysis of online algorithms/data structures. Amortized analysis is typically done via potential function method, which would be easier when the algorithm designer is allowed to tailor an algorithm towards a tentative analytical method they have in mind. However, in the context of analyzing Greedy or Splay, the algorithms are already fixed in advance (e.g. these are the algorithms that tend to work well in practice), so we have no control on the “design” part. In such cases, the state-of-the-art understanding of potential function design is much more adhoc and mostly tailored to specific cases. Indeed, there has been no systematic, efficient way known for the task of designing a potential function, when an algorithm is fixed in advance. The fact that the aforementioned conjectures have remained open for decades clearly underlines the lack of understanding on this front.

Extremal combinatorics methods (such as forbidden submatrix theory) have been used successfully in amortized analysis as an alternative to potential function design. In the context of binary search trees, such attempts were pioneered by Pettie [Pet08, Pet10] and more recently extended by [CGK<sup>+</sup>15a, CGK<sup>+</sup>15b]. Informally speaking, one can encode an execution log of Greedy as a binary matrix. It is a non-trivial fact that, since the input is restricted, the execution log of Greedy is also restricted. When the execution log is restricted, it cannot have too many 1’s in the matrix, and thus we can apply the extremal bounds from forbidden submatrix theory as a black box.

More precisely, forbidden submatrix theory is a collection of theorems of the form: Let  $\pi$  be a matrix (pattern), and  $\text{ex}(n, \pi)$  denotes the extremal bound which equals the maximum number of 1s in any  $n$ -by- $n$  0/1 matrix that *avoids* pattern  $\pi$  (a matrix  $M$  *contains* pattern  $\pi$  if it is possible to obtain  $\pi$  from  $M$  by removing rows, columns, and turning ones into zeroes; otherwise, we say that  $M$  *avoids*  $\pi$ ), see Figure 1 for illustration. Studying behavior of extremal functions for various matrices  $\pi$  have been a fruitful area of research in extremal combinatorics.

$$\pi_1 = \begin{bmatrix} \textcolor{red}{1} & 0 \\ \textcolor{red}{1} & \textcolor{red}{1} \end{bmatrix} \quad \pi_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & \textcolor{red}{1} & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & \textcolor{red}{1} & 0 & 0 & \textcolor{red}{1} \end{bmatrix}$$

Figure 1: An example that  $M$  contains pattern  $\pi_1$ , but avoids  $\pi_2$ .

Let  $X \in [n]^n$  be an input sequence. Denote by  $G_X$  the matrix that “encodes” the execution log of Greedy, that is,  $G_X(i, j) = 1$  if and only if key  $i$  is touched by Greedy at time  $j$ , implying that the number of 1s in  $G_X$  (denoted by  $|G_X|$ ) is equal to the cost of the algorithm. The connection between BSTs and the theory of forbidden matrices (see, e.g., in [CGK<sup>+</sup>15a, CGK<sup>+</sup>15b]) relies on a “reduction statement”, which says that if  $X$  avoids a pattern  $\pi$  of size  $k$ , then the Greedy matrix  $G_X$  avoids a (tensored) pattern  $\pi'$  of size  $3k$ . Therefore, existing extremal bounds can be immediately used to upper bound  $|G_X|$ . Indeed, for preorder and postorder traversals (that avoid patterns of size 3),  $G_X$  avoids a pattern of size 9. The bound of  $n2^{\alpha(n)^{O(1)}}$  follows from this generic reduction. Here, we state the known reductions.

**Lemma 1 ([CGK<sup>+</sup>15a, CGK<sup>+</sup>15b])** Let  $Q_0 = \begin{bmatrix} 1 & & 1 & & 1 \\ & 1 & & 1 & \\ & & 1 & & 1 \end{bmatrix}$ , and let  $Q_1$  and  $Q_2$  be the following matri-

ces.

$$Q_1 = \begin{bmatrix} & & 1 & & & & \\ & 1 & & 1 & & & \\ & & & & & 1 & \\ & & & & 1 & & 1 \\ & & & 1 & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}, \text{ and } Q_2 = \begin{bmatrix} & & & & 1 & & \\ & & & 1 & & 1 & \\ & & & & & & 1 \\ & & & & & & & 1 \\ & & & & & 1 & & \\ & & 1 & & & & & \\ & 1 & & 1 & & & & \\ & & & & & & & 1 \end{bmatrix}.$$

(we omit zero entries for clarity).

- If  $X$  is delete-only deque sequence, then  $G_X$  avoids  $Q_0$ . Therefore,  $|G_X| \leq \text{ex}(n, Q_0) = O(n2^{\alpha(n)})$ .
- If  $X$  is preorder traversal, then  $G_X$  avoids  $Q_1$ . Therefore,  $|G_X| \leq \text{ex}(n, Q_1) \leq n2^{\alpha(n)^{O(1)}}$ .
- If  $X$  is postorder traversal, then  $G_X$  avoids  $Q_2$ . Therefore,  $|G_X| \leq \text{ex}(n, Q_2) \leq n2^{\alpha(n)^{O(1)}}$ .

**Barrier for Improvements.** Given that Lemma 1 provide near optimal deque and traversal sequences for Greedy up to a factor of  $2^{\alpha(n)^{O(1)}}$ , it is an intriguing open question to further extend this technique to settle deque and/or traversal conjectures. As suggested by [CGK<sup>+</sup>15b], the improvement can potentially be made either finding a better pattern that is avoided by Greedy matrix, or improving the analysis of the extremal bounds for  $Q_1$  and  $Q_2$ . However, there are some inherent limitations to this approach. First, the lower bounds are known<sup>2</sup> for  $Q_1$  and  $Q_2$ . That is,  $\text{ex}(n, Q_1) = \Omega(n\alpha(n))$  and  $\text{ex}(n, Q_2) = \Omega(n\alpha(n))$ . Although they are far from the current upper bounds, these results stop us from obtaining linear upper bounds.

The second key barrier is due to a counterexample provided in [CGK<sup>+</sup>15b]. A natural way to prove that the Greedy matrix  $G(X)$  satisfies  $|G(X)| = O(n)$  is to show that  $G(X)$  avoids a constant-sized permutation pattern, and applying the upper bound from [MT04]. However, [CGK<sup>+</sup>15b] shows a family of sequences  $X$  such that  $G(X)$  contains every constant-sized permutations even when the input is delete-only deque sequence. The counterexample also suggests that it is unlikely that the Greedy matrix will avoid some *linear patterns* (i.e. pattern  $\pi$  whose extremal bound  $\text{ex}(n, \pi) = O(n)$ ).

**Our Results.** In this paper, in order to bypass the barriers, we propose to decompose the Greedy matrix  $G(X)$  into several matrices that are “easier in different ways”. More formally, we write  $G(X) = \sum_{i=1}^{\ell} M_i$  where matrices  $M_i$  are chosen based on the structures of  $X$  so that each  $M_i$  avoids a much smaller pattern  $\pi_i$  (which can be different for distinct  $i$ ). This would give the upper bound  $|G(X)| \leq \sum_{i=1}^{\ell} \text{ex}(n, \pi_i)$ . All our results follow this framework. We believe that our matrix decomposition techniques will inspire further development of amortized analysis using extremal combinatorics beyond BSTs.

- For preorder traversal input  $X$ , we have

$$|G(X)| \leq \text{ex}\left(n, \begin{bmatrix} 1 & & \\ & 1 & \end{bmatrix}\right) + \text{ex}\left(n, \begin{bmatrix} 1 & & 1 \\ & 1 & \end{bmatrix}\right) + 2 \cdot \text{ex}\left(n, \begin{bmatrix} 1 & & 1 & & 1 \\ & 1 & & 1 & \end{bmatrix}\right)$$

which implies that  $|G(X)| \leq O(n2^{\alpha(n)})$  (details in Section 4). We remark that, without the matrix de-

composition technique, the matrix  $G(X)$  itself contains pattern  $\begin{bmatrix} 1 & & 1 & & 1 \\ & 1 & & 1 & \end{bmatrix}$  (see Appendix A for a counterexample).

---

<sup>2</sup>They contain the pattern  $\begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & 1 & \end{bmatrix}$  whose extremal bound is  $\Omega(n\alpha(n))$  [FH92].

	Previous known	This paper	Remark
Preorder	$n2^{\alpha(n)^{O(1)}} [CGK^+15b]$	$O(n2^{\alpha(n)})$	Theorem 1(1)
Postorder	$n2^{\alpha(n)^{O(1)}} [CGK^+15b]$	$O(n)$	Theorem 1(2)
Deque	$m2^{\alpha(m,m+n)} [CGK^+15a]$	$O(m\alpha(n))$	Theorem 1(3)
Split	-	$O(n2^{\alpha(n)})$	Theorem 2
$k$ -Increasing	$O(nk^2) [CGK^+15b, Cib13]$	$O(\min\{nk^2, nk\alpha(n)\})$	Theorem 3

Table 1: Main results for Greedy BSTs.

- For postorder traversal input  $X$ , we have

$$|G(X)| \leq \text{ex}\left(n, \begin{bmatrix} 1 & 1 \\ & 1 \end{bmatrix}\right) + \text{ex}\left(n, \begin{bmatrix} 1 & & 1 \\ & 1 & \end{bmatrix}\right) + 2 \cdot \text{ex}\left(n, \begin{bmatrix} & 1 \\ 1 & & 1 \end{bmatrix}\right)$$

which implies that  $|G(X)| = O(n)$  (details in Section 4).

- For delete-only deque input  $X$ , we have

$$|G(X)| \leq \text{ex}\left(n, \begin{bmatrix} & 1 & & 1 \\ 1 & & 1 & \end{bmatrix}\right) + \text{ex}\left(n, \begin{bmatrix} 1 & & 1 & \\ & 1 & & 1 \end{bmatrix}\right) + O(n)$$

which implies that  $|G(X)| \leq O(n\alpha(n))$  (details in Section 3). We remark that, without the matrix de-

composition technique, the matrix  $G(X)$  itself contains pattern  $\begin{bmatrix} 1 & & 1 & \\ & 1 & & 1 \end{bmatrix}$  and  $\begin{bmatrix} & 1 & & 1 \\ 1 & & 1 & \end{bmatrix}$  (see Appendix A for a counterexample).

We summarize our results in the following.

**Theorem 1** *The following bounds hold for Greedy:*

1. Greedy searches any preorder traversal sequence with cost  $O(n2^{\alpha(n)})$ .
2. Greedy searches any postorder traversal sequence with cost  $O(n)$ .
3. Starting with any initial BST  $R$  with  $n$  keys, Greedy serves  $m$  operations of INSERTMIN, INSERTMAX, DELETEMIN, DELETEMAX with cost at most  $O(m\alpha(n))$  assuming  $m > n$ .

**Remark 1** *The sequence for preorder and postorder is a permutation sequence of length  $n$ . For deque we consider any sequence of length  $m > n$ .*

**Remark 2** *In all our results we consider an initial tree  $T$  before the execution of Greedy. If the initial tree is not given, then problems becomes much "easier" to solve. For example the cost of sequential access is  $O(n)$  with initial tree and without initial tree but the proof structures are different [CGK<sup>+</sup>15b, Fox11].*

For split conjecture, we in fact prove that the maximum cost of Greedy's splitting is at most the cost of searching a preorder traversal and therefore our traversal bound directly gives an upper bound on Greedy's serving split tree operations.

**Theorem 2 (Informal)** For  $n \in \mathbb{N}$ , let  $\text{deldeq}(n)$  be the maximum possible costs of Greedy when serving deletion-only deque operations on  $n$  keys, and  $\text{preorder}(n)$  be the maximum cost when serving preorder search. Then,

$$\text{deldeq}(n) \leq \text{split}(n) \leq \text{preorder}(n).$$

Consequently, Greedy can be used as a split tree with cost  $O(n2^{\alpha(n)})$ .

As a consequence, the traversal conjecture implies the split conjecture, which implies the delete-only deque conjecture. The implication from traversal to split conjectures is not known for Splay trees.

One can view this collection of conjectures as the dynamic optimality conjecture on restricted inputs, where the restriction on the input sequence is defined by pattern avoidance properties. Pattern avoiding problems are interesting special cases of the dynamic optimality conjecture that have shown interplay between extremal combinatorics and data structures.

We now define pattern avoidance formally. Consider any input  $X$ .<sup>3</sup> We say that  $X = (x_1, x_2, \dots, x_n)$  contains pattern  $\pi = (\pi_1, \dots, \pi_k)$  if there are indices  $i_1 < \dots < i_k$  such that the subsequence  $(x_{i_1}, x_{i_2}, \dots, x_{i_k})$  is order-isomorphic to  $\pi$ . Otherwise,  $X$  avoids  $\pi$ .

The three conjectures can be (roughly) rephrased in this language as follows. For the preorder traversal conjecture, we are given an input permutation  $X$  that avoids pattern  $(2, 3, 1)$ , and our goal is to show a binary search tree that searches this sequence with cost at most  $O(n)$ . For the postorder traversal conjecture, our goal is to search an input sequence that avoids  $(1, 3, 2)$ . For the deque conjecture, if we represent the (delete-only) input  $X$  (where  $x_t$  is the key deleted at time  $t$ ), then  $X$  avoids patterns  $(2, 3, 1)$  and  $(2, 1, 3)$ . In this way, all these conjectures deal with size-3 pattern-avoiding input classes.

Besides the small patterns, recent works have also started exploring the complexity of input classes that avoid patterns of growing sizes [CGK<sup>+</sup>15b, GG19, CGK<sup>+</sup>16]. See the thesis of Kozma [Koz16] for more detail about this connection and a broader perspective on this class of problems. Our next result shows the improvement on a pattern-avoiding input class that allow patterns to be growing in terms of  $k$ . We say that an input  $X$  is  $k$ -increasing (respectively,  $k$ -decreasing) if  $X$  avoids  $(k+1, k, k-1, \dots, 1)$  (respectively,  $(1, 2, \dots, k, k+1)$ ). Note that 1-increasing (1-decreasing) sequence corresponds to sequential sequences:  $X = (1, \dots, n)$  (or  $X = (n, \dots, 1)$ , respectively). The sequential sequence has been studied in the early days of the dynamic optimality conjecture [Tar85, Elm04] for splay and a bit more recently for Greedy [Fox11]. Note that  $k$ -increasing sequence and  $k$ -decreasing sequence are symmetric.

**Theorem 3** For  $k$ -increasing or  $k$ -decreasing input  $X$ , Greedy serves input  $X$  with cost at most  $O(\min\{nk^2, nk\alpha(n)\})$ .

Previously, the best analysis of Greedy achieves the upper bound of  $O(nk^2)$  [CGK<sup>+</sup>15b]. They showed that the greedy matrix avoids a permutation of size  $k^2$ . Furthermore, the permutation is *layered* (i.e., a concatenation of decreasing sequences into layers such that each entry of a layer is smaller than the following layers) and thus it admits  $O(nk^2)$  bounds by Theorem 6.1 of [Cib13]. Our new result improves the previous bound whenever  $k > \alpha(n)$ . Interestingly, when  $k = \omega(2^{\alpha(n)})$ , the new upper bound term  $O(nk\alpha(n))$  also improves over the best known bound even for offline optimal BST algorithm, which is  $O(nk \log k)$  [CGK<sup>+</sup>15b]. Table 1 summarizes our main results.

**A New Result in Extremal Combinatorics:** Along the way of proving  $k$ -increasing bounds for Greedy, we discover a new result in extremal combinatorics regarding the bounds of “easy” permutation patterns. A seminal result by Marcus and Tardos [MT04], show that  $\text{ex}(n, P) = O(n2^{k \log k})$  for any length- $k$  permutation matrix  $P$ . The bounds have been improved to  $\text{ex}(n, P) = n2^{O(k)}$  by [Fox13, CK17]. Furthermore, Fox [Fox13] showed (via a randomized construction) that almost all permutation matrices have the bound  $\geq n2^{\Omega((k/\log k)^{1/2})}$  and left open the following conjecture:

**Conjecture 1** If  $\pi$  is a permutation that avoids  $O(1)$ -sized pattern,  $\text{ex}(n, \pi) = n \cdot \text{poly}(|\pi|)$ .

<sup>3</sup>It was argued in [DHIP07] that one can assume w.l.o.g. that the input is a permutation.

Here, we make a partial progress by showing an approach to determine if the extremal bound of a permutation matrix  $P$  has polynomial dependence on  $k$  instead of exponential dependence. As a result, we discover a new class permutation matrices whose extremal bounds are polynomially bounded.

For any permutation  $\pi$ , denote by  $\text{dleft}(\pi)$  (abbreviation for “delete from the left”) the permutation obtained by removing the point (in the matrix form of  $\pi$ ) on the leftmost column as well as its corresponding row and column; for instance,  $\text{dleft}(1, 3, 4, 2) = \text{dleft}(2, 3, 4, 1) = (2, 3, 1)$ . Similarly, we can define  $\text{dright}(\pi)$ . We say that a length- $k$  permutation matrix  $P$  is *left-reducible* if it contains a point on one of the two corners of the first column (i.e., at coordinate  $(1, 1)$  or  $(1, k)$ ). Similarly, we say that  $P$  is *right-reducible* if it contains a point on one of the two corner of the last columns (i.e., at coordinate  $(k, 1)$  or  $(k, k)$ ).

**Definition 1** Let  $P$  be a length- $k$  permutation matrix. We say that  $P$  is reducible to a permutation matrix  $Q$ , denoted by  $P \rightarrow Q$ , if one of the followings is true

- $Q = \text{dleft}(P)$  and  $P$  is left-reducible, or
- $Q = \text{dright}(P)$  and  $P$  is right-reducible.

Furthermore, we say that  $P$  is reducible to  $Q$  in  $t$  steps, denoted by  $P \xrightarrow{t} Q$ , if there exist permutation matrices  $P_1, \dots, P_{t-1}$  such that  $P \rightarrow P_1 \rightarrow \dots \rightarrow P_{t-1} \rightarrow Q$ .

We say that a length- $k$  permutation  $P$  is  $k$ -linear if  $\text{ex}(n, m, P) = O(k(m + n))$  where  $\text{ex}(n, m, P)$  is asymmetric extremal bounds of  $m$ -by- $n$  matrix avoiding  $P$ . Similarly, we say that  $P$  is  $k$ -polynomial if  $\text{ex}(n, P) \leq nk^{O(1)}$ . Our new result is the following.

**Theorem 4** If a length- $k$  permutation  $P$  is reducible to a  $k$ -linear permutation  $Q$  in  $t$  steps, then  $\text{ex}(n, P) \leq nk^{O(t)}$ . In particular,  $P$  is  $k$ -polynomial whenever  $t = O(1)$ .

In other words, if we start with a  $k$ -linear permutation, we can add a point on one of the corners and repeat for a few times, then the resulting permutation is  $k$ -polynomial. An example of linear permutation includes an identity matrix (Theorem 7(i) of [BC21]). Another important class of linear permutation is *layered permutation*. A layered permutation is a concatenation of decreasing sequences  $S_1, \dots, S_\ell$  such that every element of  $S_i$  is smaller than all elements of  $S_{i+1}$ . Layered permutations are known to be linear in  $k$  (Theorem 1.6 of [Cib13]). We refer to [CK17] for more discussion regarding  $k$ -linear permutations.

**Further Related Work.** The “parameterized” pattern avoiding inputs, where one considers an input class whose avoided pattern has size depending on parameter  $k$ , have recently received attention (see e.g., [CGK<sup>+</sup>15b, GG19]). Research questions in this setting aim to first prove the upper bound for  $\text{OPT}(X)$  as a function of  $k$  and later show that Greedy matches this upper bound. The parameters of interest are those that generalize the classical special cases of the dynamic optimality conjecture (such as deque and pre-order traversals). Chalermsook et al. [CGK<sup>+</sup>15b] showed  $O(n2^{\alpha(n)^{O(|\pi|^2)}})$  upper bound for the cost of Greedy on inputs avoiding  $\pi$ . Goyal and Gupta proved  $O(n \log k)$  upper bound on the cost of Greedy on  $k$ -decomposable sequences [GG19] (if one allows “preprocessing”). A stronger bound that subsumes the  $k$ -decomposable bound and the dynamic finger bound [CMSS00, Col00] was shown by [IL16, BDIL16] (see discussion in [CGK<sup>+</sup>16]). Besides pattern avoidance, other BST bounds include the unified bound [BCDI07, Iac01, DS09] and the multi-finger bound [CGK<sup>+</sup>18, HIM13, DILÖ13]. The original drawback of Greedy was that, in contrast to Splay whose simplicity made it attractive for practitioners, Greedy does not admit a simple implementation in the BST model. However, due to a recent work by Kozma and Saranurak [KS19], there exists a heap data structure (called smooth heap) which matches the cost of Greedy and is implementable in practice.

**Conclusion and Open Problems.** We propose a simple idea of partitioning the execution log of Greedy into several simpler-to-analyze matrices based on the input structures and leveraging distinct patterns to upper bound each structured matrix separately. Based on this idea, we derived improved bounds for many notorious pattern avoidance conjectures (and completely settling the postorder conjecture). We view these results as a showcase of the decomposition trick, which allows us to extend/strengthen the applications of forbidden submatrices in amortized analysis of data structures. We believe that this technique would find further uses in BSTs and more broadly in amortized analysis of data structures.

**Paper Organization.** We start preliminaries including notations, definitions (the formal definition of Greedy BSTs in particular), basic facts about Greedy in Section 2. In Section 3, we start with a warm-up section providing simple proofs of sequential access theorem, and delete-only deque sequence. In Section 4, we prove Greedy bounds for preorder and postorder traversals. In Section 5, we discuss Deque sequence with insertions and deletions. In Section 6, we discuss  $k$ -decreasing sequence. In Section 7, we discuss the new result in extremal combinatorics. Finally, we discuss Split conjecture for Greedy in Section 8.

## 2 Preliminaries

**Matrix and geometry:** Let  $M$  be a binary matrix. For geometric reasons, we write matrix entries column-first, and the rows are ordered bottom-to-top, i.e. the first row is the bottom-most. Strictly speaking,  $M(i, j)$  is the value of  $i$ -th column and  $j$ -th row of  $M$ . The matrix  $M$  can be interchangeably viewed as the set of points  $\mathcal{P}(M)$  such that  $(i, j) \in \mathcal{P}(M)$  if and only if  $M(i, j) = 1$ . We abuse the notation and sometimes write  $M$  (viewing  $M$  as both the matrix and the point set corresponding to 1-entries) instead of  $\mathcal{P}(M)$ . Denote by  $|M|$  the number of 1s in  $M$ .

For point  $p \in [n] \times [m]$ , we use  $p.x$  and  $p.y$  to refer to the  $x$  and  $y$ -coordinates of  $p$  respectively. Let  $I \subseteq [n]$  and  $J \subseteq [m]$  sets of consecutive integers. We refer to  $R = I \times J$  as a rectangle. We say that matrix  $M$  is *empty* in rectangle  $R$  if  $M(i, j) = 0$  for all  $(i, j) \in R$ ; or equivalently, the rectangle  $R$  is  $M$ -empty.

Let  $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(k))$  be a permutation. We can view any permutation  $\sigma$  as a matrix  $M_\sigma$  where  $M_\sigma(\sigma(i), i) = 1$  and other entries are zero.

**Pattern avoidance:** We say that matrix  $M$  contains pattern  $P$  if  $P$  can be obtained by removing rows, columns and non-zero entries of  $M$ . If  $M$  does not contain  $P$ , we say that  $M$  avoids  $P$ . The theory of forbidden submatrices focuses on understanding the following extremal bound:  $\text{ex}(n, P)$  is defined as the maximum number of 1-entries in an  $n$ -by- $n$  matrix that avoids  $P$ .

We will use the following known bounds from [FH92, Pet11]:

**Theorem 5 ([FH92, Pet11])** •  $\text{ex}\left(n, \begin{bmatrix} & & 1 \\ 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}\right) = O(n).$

$$\bullet \text{ex}\left(n, \begin{bmatrix} & & 1 & \\ 1 & & & 1 \\ & 1 & & \\ & & 1 & \end{bmatrix}\right) = O(n\alpha(n)).$$

$$\bullet \text{ex}\left(n, \begin{bmatrix} & & & 1 \\ & 1 & & \\ 1 & & & \\ & & 1 & \end{bmatrix}\right) = O(n\alpha(n)).$$

$$\bullet \text{ex}\left(n, \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \end{bmatrix}\right) = O(n2^{\alpha(n)}).$$



**Greedy:** We consider input in the matrix  $X$ , that is,  $X(i, j) = 1$  if and only if key  $i$  is accessed at time  $j$ . Notice that each matrix row contains exactly one 1-entry. Denote by  $Y = G_T(X)$  the matrix corresponding to the execution log of Greedy on sequence  $X$  and initial tree  $T$ , that is,  $Y(i, j) = 1$  if and only if key  $i$  is touched by Greedy on initial tree  $T$  at time  $j$ . We have  $X \subseteq G_T(X)$  (Greedy always touches the input).

We explain how  $G_T(X)$  is constructed. Inputs are matrix  $X$  (one point per row) and matrix  $T$  ( $n$  rows and  $n$  columns). The columns of  $G_T(X)$  are  $[n]$  and the rows of are indexed by  $\{-(n-1), \dots, 0, 1, \dots, m\}$ . The non-positive rows are exactly by  $T$ . Greedy starts adding points into  $G_T(X)$  by processing rows  $t = 1, \dots, m$  in this order. At time  $t$ , we initialize  $S \leftarrow \emptyset$ . Let  $p$  be a point in  $X$  on  $t$ -th row. For each  $a \in [n]$ , if the rectangle  $\square_{(a, \tau(a, t)), p}$  is empty, then we add point  $(a, t)$  to  $S$ . After we process all keys  $a$ , we add points in  $S$  to  $G_T(X)$ . See Figure 2 below.

We say that points in  $X$  are accessed and points in  $G_T(X)$  are touched. Moreover, for point  $p$  in  $G_T(X)$  we say that key  $p.x$  is touched at time  $p.y$ . Throughout the paper, our statements hold for every initial tree  $T$ , hence we use  $G(X)$  instead of  $G_T(X)$ .

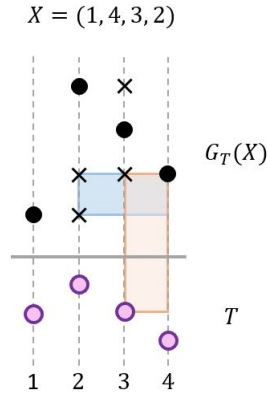


Figure 2: An example of  $G_T(X)$

We extend the pattern avoidance notation to handle multiple types of points. Let “•” denote each input point in  $X$  and “ $\times$ ” each point in  $G(X) \setminus X$ . The notation  $Y = G(X)$  contains  $\begin{matrix} a & b \\ t_2 & \begin{bmatrix} \bullet & \\ \times & \end{bmatrix} \\ t_1 \end{matrix}$  is used in the most intuitive way: Entry  $Y(a, t_2)$  contains an input, and  $Y(b, t_1)$  contains a touched point (or equivalently,  $b$  is touched at time  $t_1$ ).

We are interested in studying the pattern avoidance bound for Greedy. Define  $\text{gex}(n, P)$  as the maximum execution cost of Greedy  $G(X)$  over all permutation input  $X$  that avoids pattern  $P$  and over all initial trees  $T$ . This extremal function is a function of  $n$  and  $|P|$ .

**Multi-typed pattern avoidance:** For convenience, we extend the pattern avoidance terminology to allow points to have different types. Let  $M$  be a point set and  $\mathbb{T}$  be the set of type of points. A type function of  $M$  is a mapping  $f : M \rightarrow \mathbb{T}$ . In the matrix view, the type  $f$  assigns a value in  $\mathbb{T}$  to each non-zero entry of  $M$ .

Let  $M$  be a matrix and  $P$  a pattern. Let  $\mu$  and  $\pi$  be types of  $M$  and  $P$  respectively. We say that  $(M, \mu)$  contains  $(P, \pi)$  if and only if

- $M$  contains  $P$  or there exists a submatrix  $M'$  obtained by removing rows, columns, and points of  $M$  such that  $M' = P$ . Also,  $\pi' : M' \rightarrow \mathbb{T}$  be the type function  $\pi$  induced on  $M'$ .
- For all  $(i, j) \in P$ , we have  $\pi'(i, j) = \mu(i, j)$ .

In this paper, our types are  $\mathbb{T} = \{\times, \bullet\}$  where  $\times$  and  $\bullet$  are the touched (but non-accessed) and accessed points respectively. A Greedy matrix  $G(X)$  is naturally associated with a type function  $f : G(X) \rightarrow \mathbb{T}$

which assigns  $\times$  to points in  $G(X) \setminus X$  and  $\bullet$  to points in  $X$ . Therefore, in the statement

$$(G(X), f) \text{ avoids } \begin{bmatrix} & \times \\ \bullet & \end{bmatrix}$$

we will often omit the types and simply say  $G(X)$  instead of  $(G(X), f)$ .

If  $M$  contains a  $k$ -by- $q$  pattern  $P$ , then there exist columns  $c_1 \leq \dots \leq c_q$  and rows  $r_1 \leq \dots \leq r_k$  such that the induced submatrix of  $M$  on those contains  $P$ . In such case, we use the following notation to specify such rows and columns where the pattern appears:

$$M \text{ contains } \begin{matrix} & c_1 & \dots & c_q \\ r_k & & & \\ \vdots & & & \\ r_1 & & & \end{matrix} \begin{bmatrix} & & & \\ & P(i, j) & & \\ & & & \end{bmatrix}$$

**Input-revealing properties of Greedy:** We use a small matrix gadget that allows us to “reveal” the location of an input point in  $X$ .

**Claim 1 (Generic Capture Gadget [CGK<sup>+</sup>15b])** *If  $G(X)$  contains  $\begin{matrix} & a & b & c \\ & & \times & \\ t_1 & \times & & \times \end{matrix}$ , then the input matrix  $X$  is non-empty in the rectangle  $[a+1, c-1] \times [t_1+1, t_2]$ .*

**Claim 2 (One-sided Capture Gadget)** *If  $G(X)$  contains  $\begin{matrix} & a & b \\ & & \times \\ t_1 & \times & \end{matrix}$  or  $\begin{matrix} & a & b \\ & \bullet & \times \\ t_1 & & \end{matrix}$ , then input matrix  $X$  is non-empty in rectangle  $[a+1, \infty) \times [t_1+1, t_2]$ . This holds symmetrically for  $\begin{matrix} & a & b \\ & \times & \\ t_1 & & \times \end{matrix}$  and  $\begin{matrix} & a & b \\ \times & & \bullet \end{matrix}$ .*

**Proof:** Let  $(c, t')$  be a touched point in the rectangle  $[a+1, \infty) \times [t_1+1, t_2]$  with smallest  $t'$ . We will show that  $X$  is non-empty in the rectangle  $[a+1, \infty) \times [t_1+1, t']$ , which will imply the Claim. Assume the rectangle  $[a+1, \infty) \times [t_1+1, t']$  is input-empty. Let  $p$  be an input point at time  $t'$  such that  $p.x \leq a$ . Since  $c$  is touched at time  $t'$ , the rectangle  $[p.x, c] \times [\tau(c, t'), p.y]$  must be empty. This contradicts to the fact that  $(a, t_1) \in [p.x, c] \times [\tau(c, t'), p.y]$ .  $\square$



Figure 3: Illustrations of Claim 1 (left) and Claim 2 (right)

**Corollary 1** If  $G(X)$  contains  $\begin{matrix} & a & b \\ t_2 & & \times \\ t_1 & \times & \end{matrix}$  or  $\begin{matrix} & a & b \\ t_2 & \bullet & \times \\ t_1 & & \end{matrix}$ , then input matrix  $X$  is non-empty in rectangle

$([b, \infty) \times [t_1 + 1, t_2])$  or in  $([a + 1, \infty) \times [t_2, t_2])$ . This holds symmetrically for  $\begin{matrix} & a & b \\ t_2 & \times & \\ t_1 & & \times \end{matrix}$  and  $\begin{matrix} & a & b \\ t_2 & \times & \bullet \\ t_1 & & \end{matrix}$ .

**Proof:** Assume the input matrix  $X$  is empty in the rectangle  $([b, \infty) \times [t_1 + 1, t_2])$  and in the rectangle  $([a + 1, \infty) \times [t_2, t_2])$ . Let  $(c, t')$  be the top most input point in  $[a + 1, b - 1] \times [t_1 + 1, t_2 - 1]$ . From Claim 2,  $X$  must be non-empty in  $[a + 1, b - 1] \times [t_1 + 1, t_2 - 1]$ . This contradicts to the fact that  $(c, t')$  be the top most input point.  $\square$

**Claim 3 (Monotone Capture Gadget [CGK<sup>+</sup>15b])** If  $X$  avoids  $(1, 2, \dots, k)$  and  $G(X)$  contains

$$\begin{matrix} & a_1 & a_2 & \dots & a_{k+1} \\ t_{k+1} & \times & & & \\ \vdots & & \dots & & \\ t_2 & & & \times & \\ t_1 & & & & \times \end{matrix},$$

then the input matrix  $X$  is non-empty in the rectangle  $[a_1, a_{k+1} - 1] \times [t_1 + 1, t_{k+1}]$ .

### 3 Warm-Up

We present two warm-up proofs before proceeding to our main results: (i) a very short proof for the sequential access theorem of Fox [Fox11], and (ii) a proof when a given input sequence avoids both  $(2, 3, 1)$  and  $(2, 1, 3)$ . This is a special case of both preorder traversal and deque conjectures.

**Sequential Access Theorem:** Let  $X$  be a sequence that avoids  $(2, 1)$  (equivalently,  $X$  is the permutation  $(1, 2, \dots, n)$ ) and  $G(X)$  be the Greedy points on  $X$ . Notice that the points in  $X$  lie on the diagonal line  $x = y$ . Decompose  $G(X)$  into  $X \cup Y_L \cup Y_R$  where  $Y_L = \{q \in G(X) \mid q.y > q.x\}$  and  $Y_R = \{q \in G(X) \mid q.y < q.x\}$ . In words, the sets  $Y_L$  and  $Y_R$  are the points strictly on the left and right of the diagonal line respectively.

**Observation 1**  $Y_L$  avoids  $\begin{bmatrix} \times & \\ & \times \end{bmatrix}$ , so  $|Y_L| \leq O(n)$ .

**Proof:** Assume otherwise that  $Y_L$  contains  $\begin{matrix} & a & b \\ t_2 & \times & \\ t_1 & & \times \end{matrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b$ . Let  $p = (b, t_0)$  be the input point below  $(b, t_1)$ . Applying Corollary 1, there exists an input point  $q$  in the region  $(-\infty, b - 1] \times [t_1 + 1, t_2]$ . Notice that  $p$  and  $q$  form  $(2, 1)$  a contradiction.  $\square$

**Claim 4**  $Y_R$  avoids  $\begin{bmatrix} \times & & \times \\ & \times & \end{bmatrix}$ . Therefore,  $|Y_R| = O(n)$ .

**Proof:** Assume otherwise that  $Y_R$  contains  $\begin{matrix} & a & b & c \\ t_2 & \times & & \times \\ t_1 & & \times & \end{matrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c$ . Let  $q$  denote an input point at time  $t_2$ . Because  $(a, t_2) \in Y_R$ , we have that  $q.x < a$ . Applying Claim 2 and the fact that  $Y_R$  contains  $\begin{matrix} & b & c \\ t_2 & & \times \\ t_1 & \times & \end{matrix}$ , we would have that the rectangular region  $[b + 1, \infty) \times [t_1 + 1, t_2]$

must contain input point  $p$ , which cannot be the same point as  $q$ ; so we have  $p.y < q.y$ . The points  $p$  and  $q$  induce pattern  $(2, 1)$ , a contradiction. See Figure 4 for illustration.  $\square$



Figure 4: Illustrations of the proofs for Claim 4 (left) and Claim 5 (right)

**“Deque Access” Theorem:** Let  $X$  be an input permutation that avoids  $(2, 3, 1)$  and  $(2, 1, 3)$ . This is a special case of the deque conjecture, roughly equivalent to the case when we are only allowed to delete the minimum and maximum. In fact, as we argue later, this is a special case of preorder traversal, deque, and split conjectures. In this section, we show that  $|G(X)| = O(n\alpha(n))$ .

Let  $r \in X$  be an input point on the top row of  $X$ . We decompose  $G(X)$  into  $X \cup G_{<} \cup G_{>} \cup G_{=}$  where

- $G_{<} = \{q \in G(X) \setminus X \mid q.x < r.x\}$
- $G_{>} = \{q \in G(X) \setminus X \mid q.x > r.x\}$
- $G_{=} = \{q \in G(X) \setminus X \mid q.x = r.x\}$

To highlight the structure of this sequence, we similarly break  $X$  into  $X = \{r\} \cup X_{<} \cup X_{>}$  where  $X_{<} = \{p \in X : p.x < r.x\}$ .

**Observation 2** *The points in  $X_{<}$  and  $X_{>}$  avoid  $(2, 1)$  and  $(1, 2)$  respectively.*

This means that the input points in  $X_{<}$  form an increasing sequence, while those in  $X_{>}$  form a decreasing sequence. Intuitively, we view input points as a triangle without base, then we partition the plane into the vertical column, left side and right side of the triangle.

**Observation 3**  $|G_{=}| \leq n$ .

Next we will show that  $|G_{<}| \leq O(n\alpha(n))$ . The left-right symmetric arguments also hold for upper bounding  $|G_{>}|$ . We further decompose  $G_{<}$  into  $Y_L \cup Y_R$  where  $Y_L = \{q \in G_{<} \mid \exists s \in X \text{ with } (q.y = s.y) \wedge (q.x < s.x < r.x)\}$  and  $Y_R = G_{<} \setminus Y_L$ . In words, the sets  $Y_L$  and  $Y_R$  are the points that are “outside” and “inside” the triangle respectively.

**Observation 4**  $Y_L$  avoids  $\begin{bmatrix} \times & \\ & \times \end{bmatrix}$ , so  $|Y_L| \leq O(n)$ .

**Claim 5**  $Y_R$  avoids  $\begin{bmatrix} \times & & \times \\ & \times & \times \end{bmatrix}$ . Therefore,  $|Y_R| = O(n\alpha(n))$ .

**Proof:** Assume otherwise that  $Y_R$  contains  $\begin{matrix} & a & b & c & d \\ t_1 & \times & & \times & \\ & & \times & & \times \end{matrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c < d$ . Applying Claim 1 and the fact that  $Y_R$  contains  $\begin{matrix} & b & c & d \\ t_1 & \times & \times & \\ & & & \times \end{matrix}$ , we would have that the rectangular region  $[b+1, d-1] \times [t_1+1, t_2]$  must contain input point  $p \in X_{<}$ . Because  $(a, t_2) \in Y_R$ , we have that  $p.y < t_2$ . Applying Claim 2 and the fact that  $Y_R$  contains  $\begin{matrix} & a & p.x \\ p.y & \times & \bullet \end{matrix}$ , we can conclude that the rectangle  $(-\infty, p.x-1] \times [p.y+1, t_2]$  contains an input point  $q \in X_{<}$ . The points  $p$  and  $q$  induce pattern  $(2, 1)$ , a contradiction.  $\square$

## 4 Bounds for Input Avoiding Size-3 Patterns

There are six patterns of size three. We divide them into three different groups as follows:  $\Pi_1 = \{(1, 2, 3), (3, 2, 1)\}$ ,  $\Pi_2 = \{(2, 3, 1), (2, 1, 3)\}$  and  $\Pi_3 = \{(1, 3, 2), (3, 1, 2)\}$ . We argue that, for each such pattern class  $\Pi_i$  (for  $i = 1, 2, 3$ ), we only need to analyze the cost of Greedy on one pattern in  $\Pi_i$ . We make this claim precise as follows. For each matrix (point set)  $M$  with  $n$  columns, define the flipped matrix  $M^{flip}$  as a matrix  $M'$  obtained by reflecting  $M$  around  $y$ -axis, that is,  $M'(i, j) = M(n-i+1, j)$  for all  $i, j$ . Therefore, if we define  $P_1 = (1, 2, 3)$ ,  $P_2 = (2, 3, 1)$  and  $P_3 = (1, 3, 2)$ , we would have  $\Pi_i = \{P_i, P_i^{flip}\}$ .

**Proposition 6** For each  $i = 1, 2, 3$ , we have  $\text{gex}(n, P_i) = \text{gex}(n, P_i^{flip})$ .

**Proof:** We first prove that  $\text{gex}(n, P_i) \leq \text{gex}(n, P_i^{flip})$ . Let  $X$  and  $T$  be the input and initial tree that achieves the value  $\text{gex}(n, P_i)$ . First, notice that if  $X$  avoids  $P_i$ , then  $X^{flip}$  avoids  $P_i^{flip}$ . The following claim can be proved using the fact that Greedy is symmetric:

**Observation 5** For any  $X$  and initial tree  $T$ , we have  $(G_T(X))^{flip} = G_{T^{flip}}(X^{flip})$ .

This observation can be proved, for instance, by induction on the number of rows. Therefore,  $X^{flip}$  and  $T^{flip}$  are the inputs that prove that  $\text{gex}(n, P_i^{flip}) \geq \text{gex}(n, P_i)$ . The other direction can be argued symmetrically.  $\square$  We will use our techniques to prove the following theorems, which are the restatement of Theorem 1 (1 and 2).

**Theorem 7 (Preorder Traversal)** For  $P \in \Pi_2$ ,  $\text{gex}(n, P) = \text{gex}(n, (2, 3, 1)) = O(n2^{a(n)})$ .

**Theorem 8 (Postorder Traversal)** For  $P \in \Pi_3$ ,  $\text{gex}(n, P) = \text{gex}(n, (1, 3, 2)) = O(n)$ .

As for the input in  $\Pi_1$ , a theorem of [CGK<sup>+</sup>15b] implies that Greedy costs at most  $O(n)$ . In the following subsections we prove the above theorems.

### 4.1 Preorder Traversals

This section is devoted to proving Theorem 7. Let  $X$  be an input matrix which corresponds to a permutation that avoids  $(2, 3, 1)$ . We partition the Greedy points  $Y$  into four parts based on the location of the points with respect to the input points.

**Observation 6** For each point  $q \in Y \setminus X$ , there are (unique) input points  $p_1, p_2 \in X$  such that  $p_1.x = q.x$  and  $p_2.y = q.y$ .

Using this observation, for each such point  $q$ , if  $p_1.y < q.y$ , we say that  $q$  is a bottom point; otherwise, we say that it is a top point, so this will partition  $Y \setminus X$  into  $T \cup B$  where  $T$  and  $B$  are the top and bottom points respectively. Similarly, we define the left/right partition  $Y \setminus X = L \cup R$  where  $L$  and  $R$  are the left and right points.

These can be used to define our partition as follows:  $BR = B \cap R, BL = B \cap L, TR = T \cap R, TL = T \cap L$ .

**Lemma 2**  $BR$  avoids  $\begin{bmatrix} \times & \\ & \times \end{bmatrix}$ . Therefore,  $|BR| = O(n)$ .

**Proof:** Assume otherwise that  $BR$  contains  $\begin{bmatrix} \times & b \\ t_1 & \times \end{bmatrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b$ . Since the point  $(b, t_1)$  is a touched point in  $BR$ , there are input points  $p$  and  $q$  that are at the bottom and right of  $(b, t_1)$  respectively. From Claim 2, the region  $(-\infty, b-1] \times [t_1+1, t_2]$  must contain an input point  $r$ . The points  $p, q$  and  $r$  induce pattern  $(2, 3, 1)$  in  $X$ , a contradiction.  $\square$

**Lemma 3**  $BL$  avoids  $\begin{bmatrix} \times & & \times \\ & \times & \end{bmatrix}$ . Therefore,  $|BL| = O(n)$ .

**Proof:** Assume otherwise that  $BL$  contains  $\begin{bmatrix} a & b & c \\ t_1 & \times & \times \end{bmatrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c$ . Let  $p$  be an input point at the bottom of  $(b, t_1)$ . Let  $r$  be an input on the left of  $(a, t_2)$ . Applying Claim 2 and the fact that  $BL$  contains  $\begin{bmatrix} b & c \\ t_1 & \times \end{bmatrix}$ , we would have that the rectangular region  $[b+1, \infty) \times [t_1+1, t_2]$  must contain input point  $q$ . Notice that  $q \neq r$ , so we have that  $p, q$  and  $r$  induce pattern  $(2, 3, 1)$  in  $X$ , a contradiction.  $\square$

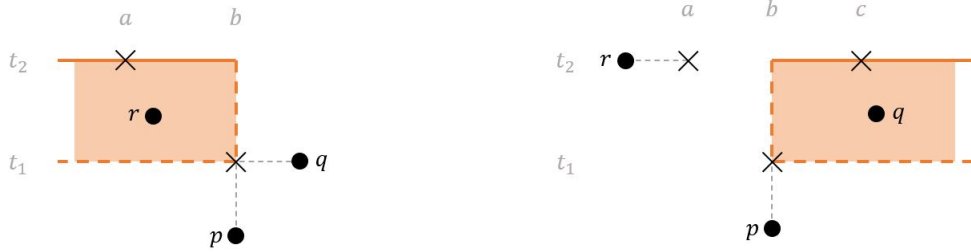


Figure 5: Illustrations of the proofs for  $BL$  (left) and  $BR$  (right) in preorder

**Lemma 4**  $TL \cup TR$  avoids  $\begin{bmatrix} \times & & \times \\ & \bullet & \end{bmatrix}$ .

**Proof:** Assume otherwise that  $TL \cup TR$  contains  $\begin{bmatrix} a & b & c \\ t_1 & \bullet & \times \end{bmatrix}$ . Let  $r$  be an input point at the top of  $(a, t_2)$ . Applying Claim 2 and the fact that  $TL \cup TR$  contains  $\begin{bmatrix} b & c \\ t_1 & \bullet \end{bmatrix}$ , we can conclude that the rect-

angle  $[b+1, \infty) \times [t_1+1, t_2]$  contains an input point  $q$ . Let  $p = (b, t_1)$ . Notice that  $p, q$  and  $r$  induce pattern  $(2, 3, 1)$ , a contradiction. See Figure 6.  $\square$

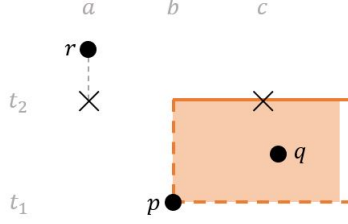


Figure 6: An illustration of the proof for  $TL \cup TR$  in preorder.

**Corollary 2** Each  $TL$  and  $TR$  avoids  $\begin{bmatrix} \times & & \times & & \times \\ & \times & & \times & \\ & & & & \end{bmatrix}$ . Therefore,  $|TL| + |TR| \leq O(n2^{\alpha(n)})$ .

**Proof:** We only present the proof for  $TL$ ; the arguments for  $TR$  are symmetric. Assume otherwise that  $TL$  contains  $\begin{bmatrix} a & b & c & d & e \\ t_2 & \times & & \times & \times \\ t_1 & & \times & & \times \end{bmatrix}$ . Applying Claim 1 to the pattern  $\begin{bmatrix} b & c & d \\ t_2 & \times & \\ t_1 & \times & \times \end{bmatrix}$ , there must be an input point  $q \in [b+1, d-1] \times [t_1+1, t_2]$ . If  $q.y < t_2$ , we would be done since it would contradict Lemma 4, so assume that  $q.y = t_2$ . Since  $a < q.x < e$ , this implies that  $(a, t_2) \in R$ , a contradiction.  $\square$

## 4.2 Postorder Traversals

This section is devoted to proving Theorem 8. Let  $X$  be a permutation that avoids  $(1, 3, 2)$ . We partition the Greedy points  $G(X)$  into four sets  $BL, BR, TL, TR$  in the same way as in the last subsection.

**Lemma 5**  $BR$  avoids  $\begin{bmatrix} & \times & \\ \times & & \times \end{bmatrix}$ . Therefore,  $|BR| \leq O(n)$ .

**Proof:** Assume otherwise that  $BR$  contains  $\begin{bmatrix} a & b & c \\ t_2 & \times & \\ t_1 & \times & \times \end{bmatrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c$ . Let  $p = (a, t_0)$  be an input at the bottom of  $(a, t_1)$ . Let  $q = (c', t_1)$  be an input on the right of  $(c, t_1)$ . Using the Claim 1, there must be an input point  $r$  in the rectangle  $[a+1, c-1] \times [t_1+1, t_2]$ . The points  $p, q$  and  $r$  induce pattern  $(1, 3, 2)$  in  $X$ , a contradiction.  $\square$

**Lemma 6**  $BL$  avoids  $\begin{bmatrix} & \times & \\ \times & & \times \end{bmatrix}$ . Therefore,  $|BL| \leq O(n)$ .

**Proof:** Assume for contradiction that  $BL$  contains  $\begin{bmatrix} a & b & c \\ t_2 & \times & \\ t_1 & \times & \times \end{bmatrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c$ . Let  $p = (a, t_0)$  be the input point at the bottom of  $(a, t_1)$ . Applying Corollary 1 with the submatrix  $\begin{bmatrix} a & c \\ t_1 & \times \\ t_0 & \bullet \end{bmatrix}$ , it follows that there exists input point  $q$  in the rectangle  $[c, \infty) \times [t_0+1, t_1]$  or

$([a+1, \infty) \times [t_1, t_1])$ . Since an input point at time  $t_1$  has to be on the left of  $a$ , This means  $q$  must be in the rectangle  $[c, \infty) \times [t_0+1, t_1-1]$ . Finally, applying Claim 1, there exists input point  $r$  in the rectangle  $[a+1, c-1] \times [t_1+1, t_2]$ . The points  $p, q$  and  $r$  induce  $(1, 3, 2)$  in  $X$ , a contradiction.  $\square$

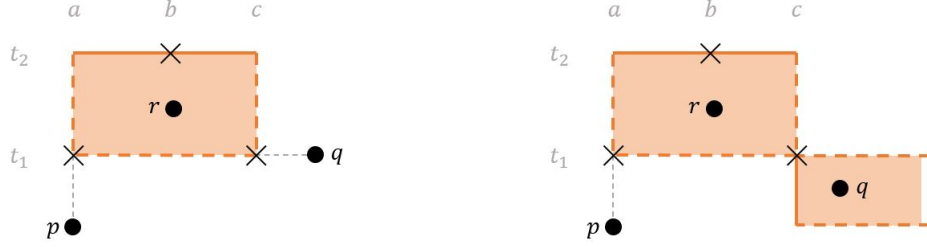


Figure 7: Illustrations of the proofs for  $BL$  (left) and  $BR$  (right) in postorder

**Lemma 7**  $TR$  avoids  $\begin{bmatrix} \times & & \times \\ & \times & \end{bmatrix}$ . Therefore,  $|TR| \leq O(n)$ .

**Proof:** Assume otherwise that  $TR$  contains  $\begin{bmatrix} a & b & c \\ t_2 & \times & \times \\ t_1 & & \times \end{bmatrix}$  for some time indices  $t_1 < t_2$  and keys  $a < b < c$ . Since  $(c, t_2)$  is in  $TR$ , there are input points  $q$  and  $r$  at the right and top of it respectively. Applying

Claim 2 to the submatrix  $\begin{bmatrix} a & b \\ t_2 & \times \\ t_1 & \times \end{bmatrix}$ , there must be an input point  $p$  in the region  $(-\infty, b] \times [t_1+1, t_2]$ . Since  $X$  is a permutation, we have that  $p.y < q.y$  (in particular,  $p \neq q$ ). The points  $p, q$  and  $r$  induce  $(1, 3, 2)$  in  $X$ , a contradiction.  $\square$

**Lemma 8**  $TL$  avoids  $\begin{bmatrix} \times & \times \\ \times & \end{bmatrix}$ . Therefore,  $|TL| \leq O(n)$ .

**Proof:** Assume otherwise that  $TL$  contains  $\begin{bmatrix} a & b \\ t_2 & \times & \times \\ t_1 & \times & \end{bmatrix}$  for time indices  $t_1 < t_2$  and keys  $a < b$ . Let  $r$  be an input at the top of  $(a, t_2)$ , and  $p$  be the input at the left of  $(a, t_1)$ . Using Claim 2 for the submatrix

$\begin{bmatrix} a & b \\ t_2 & \times & \times \\ t_1 & \times & \end{bmatrix}$ , there must exist point  $q$  in the region  $[b, \infty) \times [t_1+1, t_2]$ . The points  $p, q$  and  $r$  induce  $(1, 3, 2)$  in  $X$ , a contradiction.  $\square$

## 5 Dynamic Deque with Insertion and Deletion

In this section we prove Theorem 1(3). We use the model for dynamic update deque sequence with insertion and deletion from [CGK<sup>+</sup>15a]. An update sequence  $S$  is a set of points where each point is either inserted, deleted or accessed. For our purposes, our model only deals with insertions and deletions and no access points.

**Definition 2 (Deque Sequence)** An update sequence is a deque sequence if it only consists of INSERTMIN, INSERTMAX, DELETETMIN, DELETETMAX operations.



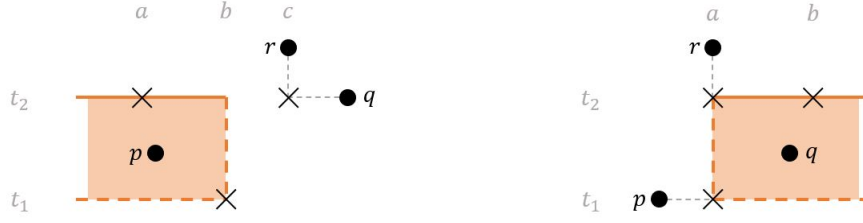


Figure 8: Illustrations of the proofs for  $TL$  (left) and  $TR$  (right) in postorder

In this model, each key can be inserted and followed by deletion at most once. In addition, it can be touched only during the time between insertion and deletion. More precisely, for any key  $p$ , let  $t_{ins}(p)$  and  $t_{del}(p)$  be an insertion and deletion time of  $p$ , respectively. If  $p$  is in an initial tree  $T$ ,  $t_{ins}(p) = 0$ . The model ensures that  $t_{ins}(p) < t_{del}(p)$ . The active time  $act(p)$  is the interval of time  $[t_{ins}(p), t_{del}(p)]$ , and  $p$  can be touched during  $act(p)$ .

Let  $Min_t$  and  $Max_t$  be the set of keys which are deleted by  $DELETEMIN$  and  $DELETEMAX$  before time  $t$ , respectively.

**Definition 3 (from [CGK<sup>+</sup>15a] Concentrated Deque Sequence)** A deque sequence is concentrated if, for any time  $t$ , the inserted element  $x$  is the minimum, then  $y < x$  for all  $y \in Min_t$ , and if  $x$  is the maximum, then  $x < y$  for all  $y \in Max_t$ .

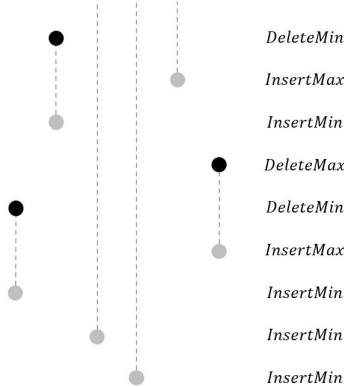


Figure 9: An example of concentrated deque sequence

In the succeeding, we will use the following lemma from [CGK<sup>+</sup>15a] to assume that the updated deque sequences we work with are always concentrated deque sequences.

**Lemma 9 (From [CGK<sup>+</sup>15a])** For any deque sequence  $S$ , there is a concentrated deque sequence  $S'$  such that the execution of any BST algorithm on  $S'$  and  $S$  have the same cost.

### 5.1 The $O(m\alpha(n))$ Bound

Let  $X$  with  $|X| = m$  be an input instance of concentrated deque with insertion and deletion where  $x_i$  is the inserted or deleted key at time  $i$ . A key  $a$  is touched (excluding deletions) by Greedy at a time  $t$  only when  $\square_{(a, \tau(a, t)), (x_t, t)}$  is empty.

**Lemma 10** Assuming  $m \geq n$ , then  $|G(X)| \leq O(m\alpha(n))$ .

Let us assume that the initial tree  $T$  has  $k_0$  active keys. We divide the execution of Greedy into phases. The first phase starts at time  $t = 1$  and lasts till time  $t = \frac{k_0}{2}$ . After the end of the first phase, let  $k_1$  be the number of active keys. Then, our second phase starts at time  $t = \frac{k_0}{2} + 1$  and lasts till time  $t = \frac{k_0}{2} + \frac{k_1}{2}$ . This process continues from one phase to another until no active keys remain or  $x_m$  is inserted or deleted. When there are no active keys at the beginning of a phase, we wait for the first key to be inserted and then begin our phase. Also, we always consider  $\lceil \frac{k}{2} \rceil$  for  $k$  active keys in any phase but for notation we will use  $\frac{k}{2}$ .

For the first phase, we see Greedy's execution on  $X$  till time  $\frac{k_0}{2}$ . At time  $\frac{k_0}{2}$ , we divide the touched point of Greedy into two parts *left* (denoted as  $\mathcal{L}_{\frac{k_0}{2}}$ ) and *right* (denoted as  $\mathcal{R}_{\frac{k_0}{2}}$ ) such that both parts contains equal number of active keys.  $\mathcal{L}_{\frac{k_0}{2}}$  contains all keys to the left of the divide and  $\mathcal{R}_{\frac{k_0}{2}}$  contains all the keys to the right of the divide. We show that the number of touched point in  $\mathcal{L}_{\frac{k_0}{2}}$  and  $\mathcal{R}_{\frac{k_0}{2}}$  is  $O(k_0\alpha(n))$ .

In general, if there are  $k_i$  active keys at the start of phase  $i$  then we show that the number of points touched by Greedy in phase  $i$  is  $O(k_i\alpha(n))$ . Summing over all phases,  $\sum_{\text{phase } i} k_i\alpha(n) = O(m\alpha(n))$ .

## 5.2 Greedy adds $O(k_i\alpha(n))$ points in phase $i$

**Lemma 11**  $\mathcal{L}_{\frac{k_i}{2}}$  avoids  $P = \begin{matrix} & a & b & c & d \\ t' & \times & & \times & \\ & & \times & & \\ t & & & \times & \end{matrix}$ .

**Proof:** Let us assume for contradiction that  $\mathcal{L}_{\frac{k_i}{2}}$  contains  $P$ . Applying Claim 1 and the fact that  $P$  contains

$\begin{matrix} & b & c & d \\ t' & & \times & \\ t & \times & & \times \end{matrix}$  we would have that the rectangular region  $[b+1, d-1] \times [t+1, t']$  must contain an input point  $q \in X$ . Since there are at most  $\frac{k_i}{2}$  INSERTMAX and DELETMAX together in phase  $i$ , point  $q$  cannot be an operation INSERTMAX or DELETMAX. Next, we will show that  $q.y \in \text{act}(b)$ , which implies that  $q$  cannot be an operation INSERTMIN or DELETMIN because  $q.x > b$ .

To show that  $q.y \in \text{act}(b)$ , it suffices to show that  $t_{\text{del}}(b) \geq t'$ . If  $t_{\text{del}}(b)$  is not in phase  $i$ , the statement trivially holds. Assume for contradiction that  $t_{\text{del}}(b)$  is in phase  $i$  and  $t_{\text{del}}(b) < t'$ . This means  $b$  gets deleted by operation DELETMIN. There are two cases:  $t_{\text{del}}(b) \in \text{act}(a)$  and  $t_{\text{del}}(b) \notin \text{act}(a)$ . In the first case, DELETMIN cannot delete  $b$  since  $a$  is active. In the second case, it means that  $t_{\text{ins}}(a) > t_{\text{del}}(b)$ , which contradicts to the fact that  $X$  is concentrated sequence.  $\square$

Similar to above lemma we can prove that  $\mathcal{R}_{\frac{k_i}{2}}$  avoids  $P' = \begin{matrix} & a & b & c & d \\ t' & & \times & & \times \\ & \times & & \times & \\ t & & & & \end{matrix}$ . This implies that the number of points in  $\mathcal{L}_{\frac{k_i}{2}}$  and  $\mathcal{R}_{\frac{k_i}{2}}$  is  $O(\frac{k_i}{2}\alpha(\frac{k_i}{2}))$  (using Theorem 5). Thus, the total number of points added by Greedy in phase  $i$  is  $O(k_i\alpha(\frac{k_i}{2}))$ . As  $n \geq k_i$ , this quantity is  $O(k_i\alpha(n))$ .

## 6 $(k-1)$ -Decreasing Sequences

This section is devoted to proving Theorem 3. We focus on  $(k-1)$ -decreasing sequences; the argument for  $(k-1)$ -increasing sequences is symmetric. The  $O(nk^2)$  bound follows from [CGK<sup>+</sup>15b] and Theorem 6.1 of [Cib13]. We focus on proving the new  $O(kn\alpha(n))$  bound, which is smaller than  $nk^2$  whenever  $k > \alpha(n)$ .

## 6.1 An $O(kn\alpha(n))$ bound

Let  $X$  be  $(k-1)$ -decreasing sequence, i.e., a sequence that avoids  $I_k = (1, 2, \dots, k)$ . For any two points  $p, q$ , we say that  $p$  dominates  $q$  (denoted by  $p \succ q$ ) if  $p.x > q.x$  and  $p.y > q.y$ . Let  $q \in G(X) \setminus X$  be a touched, non-input point. We define  $\text{chain}(q)$  to be zero if  $q$  is not dominated by any input points in  $X$ . Otherwise,  $\text{chain}(q)$  is the maximum length  $j$  such that there exists input points  $p_1, \dots, p_j \in X$  such that  $p_1 \succ \dots \succ p_j \succ q$ ; we call  $\{p_1, \dots, p_j\}$  a witness of the fact that  $\text{chain}(q) \geq j$ . Since  $X$  avoids  $(1, \dots, k)$ , we have  $\text{chain}(q) \leq k-1$  for all  $q \in Y \setminus X$ . For  $i \in \{0, 1, \dots, k-1\}$ , we define  $G_i(X) = \{q \in G(X) \setminus X : \text{chain}(q) = i\}$ . By definitions, we can partition  $G(X)$  into  $k$  parts. That is,  $G(X) \setminus X = \bigcup_{0 \leq i \leq k-1} G_i(X)$ , and thus  $|G(X)| = |X| + \sum_{0 \leq i \leq k-1} |G_i(X)|$ . So, it suffices to bound each matrix  $G_i(X)$  separately.

**Proposition 9** For all  $i$ ,  $G_i(X)$  avoids  $\begin{bmatrix} & & \times \\ \times & \bullet & \\ & & \end{bmatrix}$ .

**Proof:** Suppose that  $G_i(X)$  contains  $\begin{matrix} & a & b & c \\ t_3 & & & \times \\ t_2 & & \bullet & \\ t_1 & \times & & \end{matrix}$  for some keys  $a < b < c$  and time indices  $t_1 < t_2 < t_3$ . Denote  $p = (c, t_3)$ ,  $q = (a, t_1)$  and  $r = (b, t_2) \in X$ . Since  $p \in G_i(X)$ ,  $\text{chain}(p) = i$ , and so there are input points  $p_1 \succ \dots \succ p_i$  that dominate  $p$ , which means they dominate  $r$ . The set  $\{p_1, \dots, p_i, r\}$  is a witness that  $\text{chain}(q) \geq i+1$ , a contradiction.  $\square$

**Corollary 3** For all  $i$ ,  $G_i(X)$  avoids  $\begin{bmatrix} & & \times \\ \times & \times & \\ \times & & \times \end{bmatrix}$ . Therefore,  $|G_i(X)| = O(n\alpha(n))$ .

**Proof:** Suppose that  $G_i(X)$  contains  $\begin{matrix} & a & b & c & d \\ t_3 & & & & \times \\ t_2 & & \times & & \\ t_1 & \times & & \times & \end{matrix}$ . By Claim 1, there is an input  $(b', t'_2) \in X$

in the rectangle  $[a+1, c-1] \times [t_1+1, t_2]$ . Therefore,  $G_i(X)$  contains  $\begin{matrix} & a & b' & d \\ t_3 & & & \times \\ t'_2 & & \bullet & \\ t_1 & \times & & \end{matrix}$ , contradicting to Proposition 9.  $\square$

## 7 Extremal Combinatorics

This section is devoted to proving Theorem 4. Let  $P$  be the length- $k$  permutation in the statement. For the rest of this section, we assume that  $n$  is a power of  $4k^2$ . This will imply the theorem (removing the assumption incurs a multiplicative factor of  $O(k^2)$ ).

### 7.1 Marcus-Tardos Recurrence (Rephrased)

We explain Marcus-Tardos approach [MT04] in our language that would allow us to prove our bounds. We first introduce another extremal function  $f$  that roughly captures the maximum number of rows in a matrix, avoiding a specified pattern, that contains sufficiently many number of 1s per row.

**Definition 4** For any permutation  $\pi$  and integer  $c$ , we define  $f(c, \pi)$  to be the maximum number of rows  $r$  such that there exists a matrix  $M$  with  $r$  rows and  $c$  columns such that (i) each row has at least  $2|\pi|$  many 1's and (ii)  $M$  avoids  $\pi$ .

Notice that this definition enforces  $c \geq 2|\pi|$  with a trivial base case:

**Observation 7** For any permutation  $\pi$ , we have  $f(2|\pi|, \pi) = |\pi|$ .

Let  $\pi$  be any permutation. Let  $\pi'$  be the permutation matrix after rotating the matrix induced by  $\pi$  by 90 degree counterclockwise. Denote  $|\pi| = k$ . Marcus and Tardos relate the upper bound on  $\text{ex}(n, \pi)$  to the extremal property of  $f$  and prove that  $f(k^2, \pi) = O(k \binom{k^2}{k})$  for any permutation  $\pi$ . We rederive the bound in terms of  $f$ .

**Theorem 10**  $\text{ex}(n, \pi) = O(nk^3(f(4k^2, \pi) + f(4k^2, \pi')))$

The rest of this section is devoted to proving Theorem 10.

Their result can be rephrased in the following way:

**Lemma 12**  $\text{ex}(n, \pi) \leq (2k - 1)^2 \text{ex}(n/4k^2, \pi) + 4nk^2 \cdot f(4k^2, \pi) + 4nk^2 \cdot f(4k^2, \pi')$ .

**Proof:** The proof closely follows [MT04]. Let  $M$  be a matrix that avoids  $\pi$  and  $|M| = \text{ex}(n, \pi)$ . Let  $n' = n/(4k^2)$ . We divide the columns of  $M$  into  $n'$  groups of consecutive columns of size  $4k^2$ , and similarly we divide the rows of  $M$  into  $n'$  groups of consecutive rows of size  $4k^2$ . Let  $B_{ij}$  be the submatrix of  $M$  formed by  $i$ -th group of columns and  $j$ -th group of rows respectively. We can view  $M$  as a block matrix  $(B_{ij})_{i,j \in [n']}$ . Each block has size  $(4k^2) \times (4k^2)$ . We say that a row (or a column) is *empty* if all entries are all zero. A matrix is *empty* if all rows and columns are empty.

For each block  $B_{ij}$ , we say that  $B_{ij}$  is *wide* if it contains at least  $2k$  non-empty columns. Also, we say that  $B_{ij}$  is *tall* if it contains at least  $2k$  non-empty rows. Let  $M'$  be the  $n' \times n'$  matrix where  $M'(i, j) = 1$  if and only if  $B_{ij}$  is non-empty, not wide and not tall. Let  $T$  be the  $n' \times n'$  binary matrix where  $T(i, j) = 1$  if and only if  $B_{ij}$  is tall. Let  $W$  be the  $n' \times n'$  binary matrix where  $W(i, j) = 1$  if and only if  $B_{ij}$  is wide. Observe that each non-empty block can be tall or wide or neither, the three matrices  $M'$ ,  $T$ , and  $W$  covers all blocks  $B_{ij}$  from  $M$ . More precisely, we have  $M'(i, j) = T(i, j) = W(i, j) = 0$  if and only if  $B_{ij}$  is empty. Therefore,

$$|M| \leq (2k - 1)^2 |M'| + 16k^4 |T| + 16k^4 |W|. \quad (1)$$

The coefficient of the term  $|M'|$  is  $(2k - 1)^2$  because the number of ones in a non-wide and non-tall block is at most  $(2k - 1)^2$ . The coefficient for both  $T$  and  $W$  is  $16k^4$  because every block is  $4k^2 \times 4k^2$ . It remains to bound the number of 1's in  $M'$ ,  $W$  and  $T$ .

**The number of 1's in  $M'$ .**

**Claim 6**  $M'$  avoids  $\pi$ . Therefore,  $|M'| \leq \text{ex}(n', \pi)$ .

**Proof:** Suppose  $M'$  contains  $\pi$ . This means that there is a set of non-empty blocks  $\{B_{ij}\}$  in  $M$  such that we can form the pattern  $\pi$  by taking one 1's per block in the set. Therefore,  $M$  contains  $\pi$ , a contradiction.  $\square$

**The number of 1's in  $W$  and in  $T$ .**

**Claim 7**  $|W| \leq n' f(4k^2, \pi)$ , and  $|T| \leq n' f(4k^2, \pi')$ .

**Proof:** Since  $W$  has  $n'$  columns, it is enough to show that the number of 1's in each column of  $W$  is at most  $f(4k^2, \pi)$ . We fix an arbitrary  $j$ -th column of  $W$ . Let  $\ell$  be the number of 1's in the  $j$ -th column of  $W$  (This means there are  $\ell$  wide blocks in the  $j$ -th group of column in  $M$ .) Assume for the sake of contradiction that  $\ell > f(4k^2, \pi)$ .

For any matrix  $P$ , we define the flattening of  $P$  (denoted by  $\text{flat}(P)$ ) as a binary row-vector  $v$ , where  $v(k) = 1$  if and only if  $k$ -th column of  $P$  is non-empty. Let  $Q_j$  be the binary matrix obtained by flattening of all the blocks in the  $j$ -th column-group. More formally, matrix  $Q_j$  has  $(4k^2)$  columns and  $n'$  rows where each row  $i \in [n']$  is a flattening of  $B_{ij}$ .

**Observation 8** *If  $Q_j$  contains permutation  $\pi$ , then  $M$  contains  $\pi$ .*

Observe that the number of 1's in each row of  $Q_j$  is at least  $2k$  because  $B_{ij}$  is wide. Since  $\ell > f(4k^2, \pi)$ , and each row of  $Q_j$  has at least  $2k$  many 1's, Definition 4 implies that  $Q_j$  contains  $\pi$ , which implies that  $M$  contains  $\pi$ , a contradiction.  $\square$

We are ready to prove Theorem 10.

**Proof:** [Proof of Theorem 10] Let  $T(n) = \text{ex}(n, \pi)$ ,  $q = 4k^2$ , and  $g(q) = f_2(q, \pi) + f_2(q, \pi')$ . By Lemma 12, we have

$$\begin{aligned} T(n) &\leq (2k-1)^2 T(n/q) + nq(g(q)) \\ &\leq nq(g(q)) + \frac{(2k-1)^2}{q} nq(g(q)) + \left(\frac{(2k-1)^2}{q}\right)^2 nq(g(q)) + \dots \\ &\leq nq(g(q)) \left(\sum_{i \geq 0} \left(\frac{(2k-1)^2}{q}\right)^i\right) \\ &\leq nq(g(q))k. \end{aligned}$$

The last inequality follows since  $(2k-1)^2/q = (2k-1)^2/(4k^2) < 1$ .  $\square$

## 7.2 An Upper bound for Function $f$

For any permutation  $\pi$ , denote by  $\text{dleft}(\pi)$  (abbreviation for “delete from the left”) the permutation obtained by removing the point (in the matrix form of  $\pi$ ) on the leftmost column as well as its corresponding row and column; for instance,  $\text{dleft}(1, 3, 4, 2) = \text{dleft}(2, 3, 4, 1) = (2, 3, 1)$ . Similarly, we can define  $\text{dright}(\pi)$ .

**Theorem 11 (Reduction rules)** *Let  $\pi$  be a length- $k$  permutation whose corresponding permutation matrix contains a point on one of the two corners of the first column (i.e. at coordinate  $(1, 1)$  or  $(1, k)$ ). Then  $f(c, \pi) = O(c) \cdot f(c, \hat{\pi})$  where  $\hat{\pi}$  is any permutation such that  $\text{dleft}(\pi) = \text{dleft}(\hat{\pi})$ . Similarly, if  $\pi$  contains a point on two corners of the last column (i.e. coordinate  $(k, 1)$  or  $(k, k)$ ), then  $f(c, \pi) = O(c) \cdot f(c, \hat{\pi})$  where  $\text{dright}(\pi) = \text{dright}(\hat{\pi})$ .*

The rest of this section is devoted to proving the proof of Theorem 11. We prove the case when the permutation matrix of  $\pi$  contains  $(1, 1)$ ; other cases are symmetric.

**Lemma 13**  $f(c, \pi) \leq f(c-1, \pi) + f(c, \hat{\pi}) + 2$

**Proof:** Let  $M$  be the matrix with  $r$  rows and  $c$  columns that achieve the bound  $f(c, \pi)$ , and assume for contradiction that  $r > f(c-1, \pi) + f(c, \hat{\pi}) + 2$ .

Let  $M_1$  be the submatrix containing the bottom  $f(c-1, \pi) + 1$  rows of  $M$ . This implies that there must be a point  $p$  in the first column of  $M_1$ ; otherwise, the submatrix of  $M_1$  without its first column would give a contradiction: It has  $(c-1)$  columns,  $\alpha$  points per row and avoids  $\pi$ .

Now let  $M_2$  be the  $f(c, \hat{\pi}) + 1$  top rows of  $M$ . We know that this submatrix must contain submatrix  $\hat{P}$ ; moreover, the submatrix of  $M_2$  without its first column must contain pattern  $\text{dleft}(\hat{\pi})$ ; let us say that the points  $Q \subseteq M_2$  induce this pattern. This implies that  $\{p\} \cup Q$  induces pattern  $\pi$ , a contradiction.  $\square$

Now Theorem 11 follows by simply applying the recurrence at most  $c$  times to unfold the term  $f(c, \pi)$ :

$$f(c, \pi) \leq f(2k, \pi) + c \cdot f(c, \hat{\pi}) + 2c$$

Since  $f(2k, \pi) = k$ , we have that  $f(c, \pi) \leq k + cf(c, \hat{\pi}) + 2c \leq 4c \cdot f(c, \hat{\pi})$  as desired.

### 7.3 Proof of Theorem 4

We first state an important lemma that will be useful for proving Theorem 4. Recall that  $Q$  (in the statement of Theorem 4) is a  $k$ -linear permutation. Note that  $Q'$  denotes the rotation of  $Q$  by 90 degrees.

**Lemma 14**  $f(c, Q)$  and  $f(c, Q') \leq c$ .

**Proof:** Let  $r = f(c, Q)$ . Let  $M$  be a matrix with  $c$  columns and  $r$  rows and avoids  $Q$ . Observe that

$$2kr \leq |M| \leq \text{ex}(r, c, Q).$$

The LHS holds since we have  $2k$  ones per row in  $M$ . The RHS follows because  $M$  avoids  $Q$ . Since  $Q$  is  $k$ -linear,  $\text{ex}(r, c, Q) \leq k(r + c)$ . So,  $2kr \leq k(r + c)$ , and we have  $r \leq c$ . The proof for  $f(c, Q')$  is similar.  $\square$

We are now ready to prove Theorem 4.

**Proof:** [Proof of Theorem 4]  
we have the following.

$$\begin{aligned} \text{ex}(n, P) &= O(nk^3(f(4k^2, P) + f(4k^2, P'))) \\ &= O(nk^3(f(4k^2, Q) \cdot (4k^2)^t + f(4k^2, Q') \cdot (4k^2)^t)) \\ &= O(nk^3(4k^2)^{t+1}) \\ &= O(n4^t k^{2t+5}). \end{aligned}$$

The first inequality follows from Theorem 10. The second inequality follows from Theorem 11 and the fact that  $P$  is reducible to  $Q$  in  $t$  steps. The third inequality follows from Lemma 14.  $\square$

## 8 Split Model

In this section we prove Theorem 2. The geometric view of Greedy is invented for the purpose of search [DHI<sup>+</sup>09] and insert/delete [CGK<sup>+</sup>15a]. We will first define Greedy execution in the split model and show the relation between this model and the standard search model when  $X$  avoids some patterns. Our main theorem in this section is the following:

**Theorem 12** Let  $X \in [n]^n$  be a permutation. Then,

- If  $X$  avoids  $(1, 3, 2)$  and  $(2, 3, 1)$ , the cost of Greedy's deleting  $X$  is at most the cost of Greedy's splitting  $X$ .
- For any sequence  $X$ , there exists a sequence  $X'$  avoiding  $(2, 3, 1)$  such that Greedy's splitting  $X$  costs at most Greedy's searching  $X'$ .

**Corollary 4** For any permutation  $X$ , Greedy's splitting  $X$  costs at most  $O(n2^{\alpha(n)})$ .

**Corollary 5** For Greedy, the traversal conjecture implies the split conjecture, which implies the deque conjecture.

## 8.1 The Split Model

Let  $X = (x_1, x_2, \dots, x_n) \in [n]^n$  be a permutation input sequence of keys where  $x_i$  is split at time  $i$ . Let  $\mathcal{I}_X = \{I_{x_1}, I_{x_2}, \dots, I_{x_n}\}$  be the set of intervals defined as follows. First, we create a binary search tree  $\mathcal{T}_X$  by inserting the keys of  $X$  into an empty initial tree where  $x_i$  is inserted at time  $i$ .<sup>4</sup> Define  $I_{x_i}$  as the minimal open integer interval containing all keys in the subtree rooted at  $x_i$  in  $\mathcal{T}_X$ . Notice that  $I_{x_1} = (0, n+1)$ .

See Figure 10 for illustration. These intervals define the “active keys” for each key, that is,  $I_a$  is the interval containing active keys when key  $a$  is split.

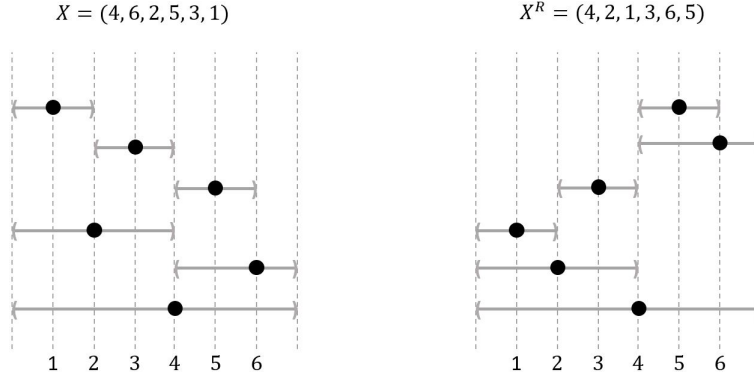


Figure 10: An example of  $X$  and  $X^R$  obtained from Algorithm 2

**Observation 9** For each  $i \in [n]$ ,  $x_i \in I_{x_i}$ .

**Observation 10** For any  $i < j$ ,  $x_i \notin I_{x_j}$ .

**Observation 11**  $\mathcal{I}_X$  is a laminar family of intervals, i.e., two intervals intersect if and only if one is completely contained in another. Furthermore, for  $j > i$ , it is either  $(I_{x_j} \subset I_{x_i})$  or  $(I_{x_j} \cap I_{x_i} = \emptyset)$ .

Let  $\tau(a, t)$  denote the last touched time of key  $a$  before time  $t$ . When  $t$  is clear from the context, we use  $\tau(a)$ . Let  $\square_{(a_1, b_1), (a_2, b_2)}$  denote the closed rectangular area defined by two points:  $(a_1, b_1)$  and  $(a_2, b_2)$ . We define the Greedy execution on input  $X$  in the split model,  $G_S(X)$ , as in Algorithm 1:

---

**Algorithm 1:** Greedy execution in the split model  $G_S(X)$

---

Given  $X$  and  $\mathcal{I}_X$   
**for**  $i \leftarrow 1$  **to**  $|X|$  **do**  
     $S = \{a \in I_{x_i} \mid \square_{(a, \tau(a)), (x_i, i)} \text{ is empty}\}$   
     $\forall a \in S$ , add point  $(a, i)$  to  $G_S(X)$

---

## 8.2 Relation to Preorder Traversals

The second part of Theorem 12 follows from the following lemmas.

<sup>4</sup>When inserting  $x_i$ , we search the current tree  $\mathcal{T}_X$  until a miss occurs, and we insert  $x_i$  at the corresponding place.

**Lemma 15** Given a permutation input sequence  $X$ , there exists a preorder permutation input sequence  $X^R$  such that  $|G_S(X^R)| = |G_S(X)|$ .

**Lemma 16** Let  $X$  be a preorder sequence. Then,  $|G_S(X)| \leq |G(X)|$ .

**Proof of Lemma 15**

For a permutation input instance  $X = (x_1, x_2, \dots, x_n)$ , we denote by  $X^R = (x_1^R, x_2^R, \dots, x_n^R)$  its rearranged permutation input instance which we construct in algorithm 2 (Figure 10).

Let  $B = (b_1, \dots, b_k)$ . Denote by  $\text{SWAP}(B, i)$  the operation that swaps  $b_i$  with  $b_{i+1}$ , that is, it returns  $B'$  which is the same as  $B$  everywhere except for  $b'_i = b_{i+1}$  and  $b'_{i+1} = b_i$ . Then  $X^R$  is obtained by iteratively applying  $\text{SWAP}$ . We argue below that  $X^R$  is a preorder traversal of binary search tree  $\mathcal{T}_X$ .

---

**Algorithm 2:** Rearrange  $X$  into  $X^R$

---

```

Function Preorder( $B = (b_1, \dots, b_k)$ ):
  while  $\exists i : (b_{i+1} < b_i) \wedge (I_{b_i} \cap I_{b_{i+1}} = \emptyset)$  do
     $B' \leftarrow \text{SWAP}(B, i)$ 
  return  $B'$ 

```

```

Function Main( $X$ ):
   $X^R \leftarrow \text{Preorder}(X)$ 
  return  $X^R$ 

```

---

**Lemma 17** Let  $B = (b_1, \dots, b_k)$ . If  $I_{b_i} \cap I_{b_{i+1}} = \emptyset$ , and  $B'$  is obtained by  $\text{SWAP}(B, i)$ . Then  $\mathcal{T}_B = \mathcal{T}_{B'}$  and hence  $\mathcal{I}_B = \mathcal{I}_{B'}$ .

In other words, this lemma proves that the BST is invariant under the swap operation.

**Proof:** Since the intervals  $I_{b_i}$  and  $I_{b_{i+1}}$  are disjoint, let  $c$  be the LCA of  $\mathcal{T}_B$  at the moment before time  $i$  (i.e. before inserting  $b_i$ ). Notice that  $b_{i+1}$  is inserted into the right subtree of  $c$ , while  $b_i$  is inserted into the left subtree of  $c$ , and the order of their insertions do not matter. Therefore,  $\mathcal{T}_B$  and  $\mathcal{T}_{B'}$  would be the same after time  $i + 1$ .  $\square$

**Claim 8**  $X^R$  is preorder sequence. In particular, it is a preorder traversal of  $\mathcal{T}_X$ .

**Proof:** First, we argue that  $X^R$  avoids  $(2, 3, 1)$ . Assume otherwise that it contains  $i < j < k$  such that  $x_i^R, x_j^R, x_k^R$  induce  $(2, 3, 1)$ , so we must have that  $b_j > b_i > b_k$ . Let  $j' : j \leq j' < k$  be the minimum integer such that  $b_{j'} > b_{j'+1}$  (notice that such  $j'$  must exist). Notice that  $I_{j'}$  ends before  $b_i$  while  $I_{j'+1}$  starts after  $b_i$  so they are disjoint. This implies that the swap would have been applied at  $j'$ , a contradiction. Since  $X^R$  is a preorder permutation, it must be a preorder permutation of  $\mathcal{T}_{X^R}$ . From Lemma 17, we have  $\mathcal{T}_X = \mathcal{T}_{X^R}$ .  $\square$

**Lemma 18**  $|G_S(B')| = |G_S(B)|$ .

**Proof:** Let  $M[t]$  denote the row  $t$  of matrix  $M$  (recall that this paper start indexing from the bottom most row). It is easy to see that  $G_S(B')[t] = G_S(B)[t]$  when  $t < i$  because both sequences are similar up to time  $i - 1$ . Next, we claim that  $G_S(B')[i] = G_S(B)[i + 1]$ . This is because  $I_{b'_i} = I_{b_{i+1}}$  and  $G_S(B)(p, i) = 0$  for all  $p \in I_{b_i}$ . So, for any key  $a \in [n]$ ,  $\square_{(a, \tau(a)), (b'_i, i)}$  is empty in  $G_S(B')$  if and only if  $\square_{(a, \tau(a)), (b_{i+1}, i+1)}$  is empty in  $G_S(B)$ . Similar argument holds for  $G_S(B')[i + 1] = G_S(B)[i]$ .



Lastly, we claim that  $G_S(B')[t] = G_S(B)[t]$  when  $t > i + 1$ . Let  $r$  be the first time after  $i + 1$  such that  $G_S(B')[r] \neq G_S(B)[r]$ . We claim that, for any key  $a \in [n]$ ,  $\square_{(a,\tau(a)),(b'_r,r)}$  is empty in  $G_S(B')$  if and only if  $\square_{(a,\tau(a)),(b_r,r)}$  is empty in  $G_S(B)$ . There are 4 cases:

1. if  $\tau(a) > i + 1$  in  $G_S(B)$ , this is trivial by our assumption.
2. if  $\tau(a) = i + 1$  in  $G_S(B)$ ,  $\square_{(a,i+1),(b_r,r)}$  is empty in  $G_S(B)$  if and only if  $\square_{(a,i),(b'_r,r)}$  is empty in  $G_S(B')$ . This is because  $G_S(B)(p, i) = 0$  for all  $p \in I_{b'_{i+1}}$ .
3. when  $\tau(a) = i$  in  $G_S(B)$ , this is symmetric to the above case.
4. when  $\tau(a) < i$  in  $G_S(B)$ . Notice that the only difference between  $G_S(B)$  and  $G_S(B')$  before time  $r$  are in rows  $i$  and  $i + 1$ . One can view  $\square_{(a,\tau(a)),(b_r,r)}$  as a set of consecutive columns. Since  $G_S(B')[i] = G_S(B)[i + 1]$  and  $G_S(B')[i + 1] = G_S(B)[i]$ , this means  $\square_{(a,\tau(a)),(b'_r,r)}$  is empty in  $G_S(B')$  if and only if  $\square_{(a,\tau(a)),(b_r,r)}$  is empty in  $G_S(B)$ .

□

### Proof of Lemma 16



Figure 11: Illustrations of the proofs of Lemma 19 (left) and Lemma 20 (right)

**Lemma 19** Let  $X$  be a preorder sequence. For each  $q \in X$ , there is no point  $r \in X$  such that  $r.x < \text{left}(I_q)$  and  $r.y > q.y$ .

**Proof:** If  $\text{left}(I_q) \leq 1$ , the lemma trivially holds. Consider the case where  $\text{left}(I_q) > 1$ . Assuming such  $r$  exists. By interval construction, there is an input point  $p$  such that  $p.x = \text{left}(I_q)$  and  $p.y < q.y$ . This means points  $p, q$  and  $r$  form  $(2, 3, 1)$  in  $X$ . Contradiction. □

**Lemma 20** Let  $X$  be a preorder sequence. For each  $r \in X$ , there is no point  $c \in G(X)$  such that  $c.x > \text{right}(I_r)$  and  $c.y = r.y$ .

**Proof:** If  $\text{right}(I_r) \geq n$ , the lemma trivially holds. Consider the case where  $\text{right}(I_r) < n$ . By interval construction, there is an input point  $p$  such that  $p.x = \text{right}(I_r)$  and  $p.y < r.y$ . If  $p.y = r.y - 1$ , there is no such touch point  $c$  because  $\square_{(c,x,\tau(c.x)),(r.x,r.y)}$  must contain  $p$ . If  $p.y < r.y - 1$ , assuming there is such touch point  $c$ . Using Claim 2 with  $p$  and  $c$ , we have that the rectangle  $[p.x + 1, \infty) \times [p.y + 1, r.y - 1]$  must contain some input  $q$  ( $q.y \neq r.y$  because  $X$  is permutation). This means points  $p, q$  and  $r$  form  $(2, 3, 1)$  in  $X$ . Contradiction. □

For  $i \in [n]$ , let  $G^{(i)}(X)$  and  $G_S^{(i)}(X)$  denote a set of points in row  $i$  of  $G(X)$  and  $G_S(X)$ , respectively.

**Lemma 21** Let  $X$  be a preorder sequence. For  $i \in [n]$ ,  $G^{(i)}(X) \cap I_{x_i} = G_S^{(i)}(X) \cap I_{x_i}$ .

**Proof:** Let  $j$  be the first time that  $G^{(j)}(X) \cap I_{x_j} \neq G_S^{(j)}(X) \cap I_{x_j}$ . This means there exists a point  $c \in G(X) \setminus G_S(X)$  such that  $c.x \in I_{x_j}$  and  $c.y < j$ . Let  $x_t$  be an input point at time  $c.y$ . From Observation 11 and the fact that  $c \notin I_{x_t}$ , we have that  $I_{x_t} \cap I_{x_j} = \emptyset$ . We divide into two cases: 1)  $\text{right}(I_{x_t}) \leq \text{left}(I_{x_j})$  and 2)  $\text{right}(I_{x_j}) \leq \text{left}(I_{x_t})$ . In the first case,  $x_t$  and  $c$  contradict Lemma 20. In the second case,  $x_t$  and  $x_j$  contradict Lemma 19.  $\square$

## References

- [BC21] Richard A. Brualdi and Lei Cao. Pattern-avoiding  $(0, 1)$ -matrices and bases of permutation matrices. *Discret. Appl. Math.*, 304:196–211, 2021.
- [BCDI07] Mihai Bădoiu, Richard Cole, Erik D Demaine, and John Iacono. A unified access bound on comparison-based dynamic dictionaries. *Theoretical Computer Science*, 382(2):86–96, 2007.
- [BDIL16] Prosenjit Bose, Karim Douïeb, John Iacono, and Stefan Langerman. The power and limitations of static binary search trees with lazy finger. *Algorithmica*, 76(4):1264–1275, 2016.
- [CGK<sup>+</sup>15a] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Greedy is an almost optimal deque. *CoRR*, abs/1506.08319, 2015.
- [CGK<sup>+</sup>15b] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Pattern-avoiding access in binary search trees. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 410–423. IEEE, 2015.
- [CGK<sup>+</sup>16] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. The landscape of bounds for binary search trees. *arXiv preprint arXiv:1603.04892*, 2016.
- [CGK<sup>+</sup>18] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Multi-finger binary search trees. *arXiv preprint arXiv:1809.01759*, 2018.
- [CH93] Ranjan Chaudhuri and Hartmut Höft. Splaying a search tree in preorder takes linear time. *SIGACT News*, 24(2):88–93, 1993.
- [Cib13] Josef Cibulka. Extremal combinatorics of matrices, sequences and sets of permutations. 2013.
- [CK17] Josef Cibulka and Jan Kyncl. Better upper bounds on the Füredi-Hajnal limits of permutations. In *SODA*, pages 2280–2293. SIAM, 2017.
- [CMSS00] Richard Cole, Bud Mishra, Jeanette Schmidt, and Alan Siegel. On the dynamic finger conjecture for splay trees. part i: Splay sorting log  $n$ -block sequences. *SIAM Journal on Computing*, 30(1):1–43, 2000.
- [Col00] Richard Cole. On the dynamic finger conjecture for splay trees. part ii: The proof. *SIAM Journal on Computing*, 30(1):44–85, 2000.
- [DHI<sup>+</sup>09] Erik D Demaine, Dion Harmon, John Iacono, Daniel Kane, and Mihai Patrăşcu. The geometry of binary search trees. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 496–505. SIAM, 2009.
- [DHIP07] Erik D Demaine, Dion Harmon, John Iacono, and Mihai Patrăşcu. Dynamic optimality—almost. *SIAM Journal on Computing*, 37(1):240–251, 2007.
- [DILÖ13] Erik D Demaine, John Iacono, Stefan Langerman, and Özgür Özkan. Combining binary search trees. In *International Colloquium on Automata, Languages, and Programming*, pages 388–399. Springer, 2013.

- [DS09] Jonathan C Derryberry and Daniel D Sleator. Skip-splay: Toward achieving the unified bound in the bst model. In *Workshop on Algorithms and Data Structures*, pages 194–205. Springer, 2009.
- [Elm04] Amr Elmasry. On the sequential access theorem and deque conjecture for splay trees. *Theoretical Computer Science*, 314(3):459–466, 2004.
- [FH92] Zoltán Füredi and Péter Hajnal. Davenport-schinzel theory of matrices. *Discret. Math.*, 103(3):233–251, 1992.
- [Fox11] Kyle Fox. Upper bounds for maximally greedy binary search trees. In *Workshop on Algorithms and Data Structures*, pages 411–422. Springer, 2011.
- [Fox13] Jacob Fox. Stanley-wilf limits are typically exponential. *arXiv preprint arXiv:1310.8378*, 2013.
- [GG19] Navin Goyal and Manoj Gupta. Better analysis of binary search tree on decomposable sequences. *Theoretical Computer Science*, 776:19–42, 2019.
- [HIM13] John Howat, John Iacono, and Pat Morin. The fresh-finger property. *arXiv preprint arXiv:1302.6914*, 2013.
- [Iac01] John Iacono. Alternatives to splay trees with  $o(\log n)$  worst-case access times. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 516–522, 2001.
- [Iac13] John Iacono. In pursuit of the dynamic optimality conjecture. In *Space-Efficient Data Structures, Streams, and Algorithms*, pages 236–250. Springer, 2013.
- [IL16] John Iacono and Stefan Langerman. Weighted dynamic finger in binary search trees. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 672–691. SIAM, 2016.
- [Koz16] László Kozma. Binary search trees, rectangles and patterns. 2016.
- [KS19] László Kozma and Thatchaphol Saranurak. Smooth heaps and a dual view of self-adjusting data structures. *SIAM Journal on Computing*, 49(5):STOC18–45, 2019.
- [LT19] Caleb C Levy and Robert E Tarjan. Splaying preorders and postorders. In *Workshop on Algorithms and Data Structures*, pages 510–522. Springer, 2019.
- [Luc88] Joan Marie Lucas. *Canonical forms for competitive binary search tree algorithms*. Rutgers University, Department of Computer Science, Laboratory for Computer ..., 1988.
- [Luc92] Joan M Lucas. On the competitiveness of splay trees: Relations to the union-find problem. *On-line Algorithms*, 7:95–124, 1992.
- [MT04] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the stanley–wilf conjecture. *Journal of Combinatorial Theory, Series A*, 107(1):153–160, 2004.
- [Mun00] J Ian Munro. On the competitiveness of linear search. In *European Symposium on Algorithms*, pages 338–345. Springer, 2000.
- [Pet08] Seth Pettie. Splay trees, davenport-schinzel sequences, and the deque conjecture. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1115–1124, 2008.
- [Pet10] Seth Pettie. Applications of forbidden 0–1 matrices to search tree and path compression-based data structures. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete algorithms*, pages 1457–1467. SIAM, 2010.
- [Pet11] Seth Pettie. Generalized davenport-schinzel sequences and their 0-1 matrix counterparts. *J. Comb. Theory, Ser. A*, 118(6):1863–1895, 2011.

- [ST85] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *Journal of the ACM (JACM)*, 32(3):652–686, 1985.
- [Sun92] Rajamani Sundar. On the deque conjecture for the splay algorithm. *Combinatorica*, 12(1):95–124, 1992.
- [Tar85] Robert Endre Tarjan. Sequential access in splay trees takes linear time. *Combinatorica*, 5(4):367–378, 1985.

## A Counter Examples

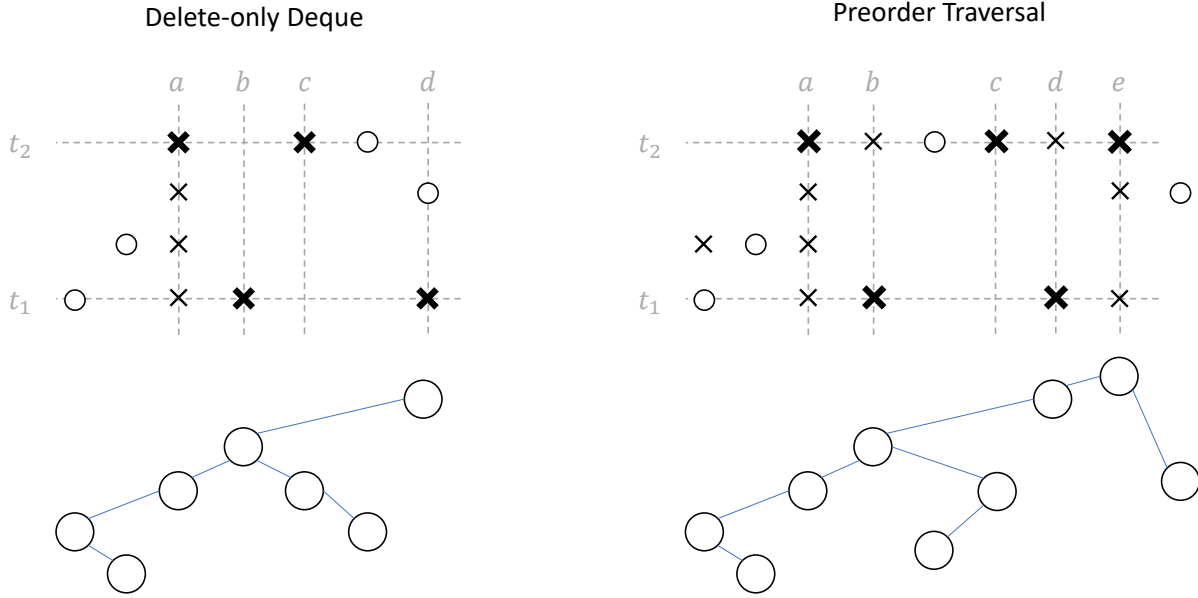


Figure 12: (Left) Greedy on Delete-only Deque Sequences with initial tree below time  $t_1$ . (Right) Greedy on Preorder Traversal with initial tree below time  $t_1$ .

The counter examples are shown in Figure 12. The definition of Greedy is discussed in Section 2. In

Figure 12 (left), the points  $(a, t_2), (b, t_1), (c, t_2)$  and  $(d, t_1)$  form the pattern  $\begin{bmatrix} 1 & & 1 & \\ & 1 & & 1 \end{bmatrix}$ . In Figure 12

(right), the points  $(a, t_2), (b, t_1), (c, t_2), (d, t_1)$  and  $(e, t_2)$  form the pattern  $\begin{bmatrix} 1 & & 1 & & 1 \\ & 1 & & 1 & \end{bmatrix}$ .