Parameterizing the Hardness of Binary Search Tree Access Sequences by Inversion Counts

Meng He* Richard Peng† Yinzhan Xu‡

Abstract

We study a new way of measuring the expected performance of various binary search tree algorithms that is between the worst and the average case. Our starting point is the correspondence between binary search trees and insertion sequences, and we will measure the difficulty of such sequences based on inversion counts. This measure naturally interpolates between random and sequential insertion orders. We show that if the tree is randomly picked from all trees built upon insertion length n permutations with t inversions, the height of the tree can be bounded by $O(n^2 \log n / \min\{t, \binom{n}{2} - t\})$.

1 Introduction

Binary search trees are fundamental structures widely used in computer science. They maintain a dynamic set of keys to support efficient search, insertion and deletion operations. These operations can be done in logarithmic time using balanced binary search trees such as red-black trees and AVL trees. Experimental studies have also shown that these balancing algorithms also give better performances over typical real-world data sets from system software products [9].

Despite this superior performance of balanced binary search trees in both theory and practice, it is also the case that on randomly generated access sequences plain unbalanced binary trees often perform on par or better than balanced ones [9]. This phenomenon can be explained theoretically: inserting a sequence of keys generated uniformly randomly leads to a tree whose height, or maximum distance from root to a node, is logarithmic with high probability [4]. Thus, under such settings, a binary search tree performs operations in logarithmic time with high probability, without the overhead of the balancing operations. In concurrent systems, uniformly random key sequences favor plain binary search trees even more, as balancing operations may incur more contention between threads [6].

What, then, distinguishes the access sequences arising from systems software from random sequences? In particular, what properties of such sequences favors tree balancing mechanism over plain unbalanced trees? An important factor, as observed by Pfaff [9], is that keys in the access sequences from system software are often partially sorted. For instance, in a Linux operation system, a process may frequently make system calls to map and unmap disk blocks into its virtual memory area. These system calls often access a sequence of consecutive virtual memory addresses or disk blocks, and each such access sequence corresponds to a sorted subsequence of requests to the search tree. This form of *presortedness* is exactly the worst case behavior for binary search tree without balancing mechanisms. In the extreme case of a completely sorted sequence, the height of the tree would grow linearly in the number of keys inserted.

In this paper, we introduce a theoretical model for measuring the difficulties of insertion sequences that allow us to gain further insights into the performance of binary search trees on partially sorted sequences. The problem can be modeled as follows: Following works on random access sequences such as [1, 11, 7] [1] we consider a permutation, σ, of the keys $\{1, 2, \ldots, n\}$ being inserted one by one in from σ_1 to σ_n. However, we introduce another parameter to measure the difficulty of such a permutation σ, which is the total number of *inversion pairs* in σ. Here an inversion pair, or simply inversion, is defined to be a pair of indices i and j such that $i < j$ but $\sigma_i > \sigma_j$. We will interpolate between random and sequential insertion orders by only considering sequences with a fixed number of inversions, t. Specifically, in this paper we investigate the height of a binary tree generated from a uniformly chosen σ with exactly t inversions.

The use of inversion counts in the analysis of algorithms and data structures has a long history. It is the key potential function in the analysis of the competitiveness of move-to-front for accessing lists

*mhe@cs.dal.ca. Supported by NSERC.

†rpeng@cc.gatech.edu. Partially supported by the National Science Foundation under Grant No. 1637566.

‡xyzhan@mit.edu

[1] Our main reference for existing works on expected behaviors of binary search trees under random accesses are from Section 3.4 of a textbook by Goodrich and Tamassia [4].

by Sleator and Tarjan [12]. Many adaptive sorting algorithms [2, 10, 3] also use inversion counts as the standard measurement of presortedness. Recently, He and Li [6] chose to generate permutations with different inversion counts as operation sequences to simulate real-world data in their experimental studies of concurrent binary search trees.

For this new problem that we proposed, we show an upper bound on the expected height of a binary search tree based on the inversion count of distribution that generates the insertion sequence. More specifically, we showed that, the expected height of a binary search tree generated from a permutation of n distinct keys uniformly chosen with t inversions is $O\left(\frac{n^2 \log n}{\min\left\{t, \binom{n}{2}-t\right\}}\right)$, where $0 < t < \binom{n}{2}$. Furthermore, the height of such a tree is within this bound with high probability. An interesting special case of this result is that, if a binary search tree is randomly picked from all trees built upon insertion sequences with cn^2 inversions for any constant $c \in (0, 1/2)$, the expected height of the tree is $O(\log n)$, and the actual tree height is within this bound with high probability. To achieve these results, we express the expected depth of a node with coefficients of a generating function. Then we utilize the log-concavity of terms of the generating function to both bound such coefficients, and to derive necessary negative correlation bounds to give with-high-probability results.

The parameterization of pattern difficulties in terms of t, inversion count, has much to be desired in terms of capturing the difficulty of access patterns. Patterns as simple as accessing all odd indices before all even indicies,

$$1, 3, 5, \dots (2k-1), 2, 4, 6, \dots (2k),$$

benefit greatly from balancing algorithms. We believe analyzing expected behaviors of interleaving several sequences, and investigating other parameters that better correspond to the usefulness of balancing mechanisms are both interesting directions for further work.

The paper is organized as follows: Section 2 provides background on generating functions, the bound on expected height is in Section 3, and its extension to a with high probability bound is in Section 4.

2 Background and Preliminaries

In this section we formalize our model for modeling random binary search trees, and discuss the primary tool that we use in our analyses, generating functions. In this paper, we will frequently use the term "height of a tree" to denote the distance between the root node to the farthest leaf node and the term "depth of a node" to denote the distance between the root node to this node. It is worthwhile to note that the depth of a deepest node in a tree is the same as the height of the tree, so we will use depth of nodes to bound the height of the tree.

2.1 Generating Functions

Our analyses are heavily based on generating functions, which are ways to represent sequences as coefficients of polynomials. For a polynomial $f(x)$, we will use $[x^c]f(x)$ to denote the coefficient of x^c in f. That is, for a generating function $f(x)$ with coefficients $f(x) = \sum_k a_k x^k$, we have $[x^c]f(x) = a_c$.

Most of the generating functions that we use will be composed from a sum of singleton terms. We will use $f_{\text{SUM}(i)}(x)$ as a shorthand to denote the sum of terms up to $i-1$,

$$f_{\text{SUM}(i)}(x) \overset{\text{def}}{=} \sum_{0 \le j < i} x^j,$$

as well as the 'product' of these terms over a set of indices I:

$$f_{\text{PROD}(I)}(x) \overset{\text{def}}{=} \prod_{i \in I} f_{\text{SUM}(i)}(x).$$

The usefulness of these product terms is immediate from the following Lemma.

LEMMA 2.1. *Let $[n]$ be the set of integers $1 \dots n$. The number of permutations of $1 \dots n$ that have t inversions is $[x^t]f_{\text{PROD}([n])}(x)$.*

Proof. We will show by induction on n that the coefficients of $f_{\text{PROD}(n)}(x)$ encodes the right counts for all values of t. For the base case of $n = 1$, there is only one possible permutation, which has no inversion, and $f_{\text{PROD}([1])}(x) = 1$.

For the inductive hypothesis, assume $f_{\text{PROD}([n-1])}(x)$ encodes all the inversion counts for permutations on $1 \dots n-1$. Now consider any t, and the number of permutations on $1 \dots n$ with exactly t inversions.

In a permutation, the value of the last element has n choices. If the last element is $k \in [1, n]$, then it contributes $k-1$ inversion pairs with elements before it. These first $n-1$ elements can in turn be viewed as a permutation of $1 \dots n-1$, so by the inductive hypothesis, we get that the number of such permutations is

$$\sum_{k=1}^{n} \left[x^{t-k+1}\right] f_{\text{PROD}([n-1])}(x),$$

which is precisely the coefficient of x^t in $f_{\text{PROD}([n])}(x)$.

2.2 Log-Concavity of Coefficients

Our proofs will rely extensively on the log-concavity of the coefficients in $f_{\textsc{Prod}(I)}(x)$. These claims are all standard in the generating function literature [13], but we include proofs for them in Appendix A for completeness.

A positive sequence $a_0, a_1, \ldots, a_d$ is log-concave if

$$\frac{a_k}{a_{k-1}} \geq \frac{a_{k+1}}{a_k}$$

or equivalently

$$a_k^2 \geq a_{k-1}a_{k+1}$$

for $0 < k < d$.

The following lemma is a property of log-concave sequences that we will extensively use in this paper.

Lemma 2.2. *Let $a_0, \ldots, a_d$ be a positive log-concave sequence. For any $0 \leq k_1 \leq k_2 \leq k_3 \leq k_4 \leq d$ such that $k_1 + k_4 = k_2 + k_3$, we have*

$$a_{k_1}a_{k_4} \leq a_{k_2}a_{k_3}$$

Furthermore, we can also establish that the coefficients of $f_{\textsc{Prod}([I])}(x)$ are log-concave. The following lemma is a direct consequence of the fact that log-concavity of generating function coefficients is preserved under multiplication (see e.g. Proposition 2 of [13]).

Lemma 2.3. *For any set of integers I, the coefficients of $f_{\textsc{Prod}(I)}(x)$ are log-concave. That is, if $f_{\textsc{Prod}(I)}(x))$ has degree d, then for any $0 < k < d$ we have*

$$\frac{\left[x^k\right] f_{\textsc{Prod}(I)}(x)}{\left[x^{k-1}\right] f_{\textsc{Prod}(I)}(x)} \geq \frac{\left[x^{k+1}\right] f_{\textsc{Prod}(I)}(x)}{\left[x^k\right] f_{\textsc{Prod}(I)}(x)},$$

3 Computing Expected Depth Using Generating Functions

In this section we present a proof on bounding the expected depth of a single key in a binary search tree generated from a permutation $\sigma_1 \ldots \sigma_n$ chosen uniformly from permutations with t inversions.

Critical to our calculation is the product function with some index removed. Specifically, we define

$$\begin{aligned} f_{\textsc{Prod}([n]\setminus i)}(x) &= \frac{f_{\textsc{Prod}([n])}(x)}{f_{\textsc{Sum}(i)}(x)} \\ &= \prod_{1 \leq j \leq n, j \neq i} f_{\textsc{Sum}(j)}(x). \end{aligned}$$

The importance of this function is due to the following identity

Corollary 3.1. *For any pair of keys $i < j$, and any t, the number of permutations $\sigma_1 \ldots \sigma_n$ with t inversions which leads to trees where i is an ancestor of j is exactly*

$$\left[x^t\right] f_{\textsc{Prod}([n]\setminus(j-i+1))}(x).$$

It is a consequence of the more general result in Lemma 4.2 in Section 4, which gives bound on the probability of multiple keys being a key's ancestor simultaneously. We defer its proof to that section. Also, if we change any key δ_i in a permutation to $n - \delta_i + 1$, the ancestor-descendant relationship is an invariant in this change. Therefore, when $i > j$, we can infer that the number of trees where i is an ancestor of j is exactly

$$\left[x^{\binom{n}{2}-t}\right] f_{\textsc{Prod}([n]\setminus(i-j+1))}(x).$$

This means we can immediately obtain the expected depth of a key j via linearity for expectation ($inv(\sigma)$ denotes the number of inversions in σ):

$$\begin{aligned} &\mathbb{E}_{\sigma:inv(\sigma)=t}\left[h(j)\right] \\ &= \sum_{1 \leq i \leq n} \Pr_{\sigma:inv(\sigma)=t}\left[i \text{ is ancestor of } j\right] \\ &\leq 1 + \sum_{1 \leq i \leq n} \frac{\left[x^t\right] f_{\textsc{Prod}([n]\setminus i)}(x) + \left[x^{\binom{n}{2}-t}\right] f_{\textsc{Prod}([n]\setminus i)}(x)}{\left[x^t\right] f_{\textsc{Prod}([n])}(x)}. \end{aligned} \tag{3.1}$$

We will bound individual terms in the summation, $[x^t]f_{\textsc{Prod}([n]\setminus i)}(x)/[x^t]f_{\textsc{Prod}([n])}(x)$. If we express $f_{\textsc{Prod}([n])}(x)$ as a summation over coefficients of $f_{\textsc{Prod}([n]\setminus i)}(x)$, this turns into

$$\frac{\left[x^t\right] f_{\textsc{Prod}([n]\setminus i)}(x)}{\left[x^t\right] f_{\textsc{Prod}([n])}(x)} = \frac{\left[x^t\right] f_{\textsc{Prod}([n]\setminus i)}(x)}{\sum_{k=0}^{i-1}\left[x^{t-k}\right] f_{\textsc{Prod}([n]\setminus i)}(x)}.$$

This means it's in turn sufficient to bound $[x^t]f_{\textsc{Prod}([n]\setminus i)}(x)/[x^{t-k}]f_{\textsc{Prod}([n]\setminus i)}(x)$ for some value of k, or equivalently bounding the value of $[x^t]f_{\textsc{Prod}([n]\setminus i)}(x)$ for all t in some interval. To do so, we use the log-concavity of the coefficients of the terms of $f_{\textsc{Prod}([n]\setminus i)}(x)$ as stated in Section 2.2. This condition implies that:

1. $[x^t]f_{\textsc{Prod}([n]\setminus i)}(x)$ increases from the boundary (0) to the middle, and decreases after that.

2. The relative rate of increase decreases as t moves from beginning to the end, and moves from positive to negative at the middle.

Combining these with a bound on the maximum value of $[x^t]f_{\textsc{Prod}([n]\setminus i)}(x)$ allows us to bound the maximum rate of change in some range. This bound then in turn gives a bound on the ratio between coefficients in that range.

Lemma 3.1. *Let $a_0 \ldots a_d$ be any sequence of log-concave numbers in the range $[1, n!]$. For any indices k and l satisfying*

$$0 < k < l < d,$$

and

$$l - k \leq \frac{\min\{k, d - l\}}{n \ln n},$$

we have

$$\frac{1}{e} \leq \frac{a_l}{a_k} \leq e$$

Proof. We define the rate of change as

$$\beta_i \stackrel{\text{def}}{=} \frac{a_{i+1}}{a_i}.$$

The concavity assumption implies that β is a monotonically non-increasing sequence. Applying this then gives for any k,

$$\frac{a_k}{a_0} = \prod_{i=0}^{k-1} \beta_i \geq \beta_k^k.$$

On the other hand, the assumption of $a_k \leq n!$ and $a_0 \geq 1$ gives $\beta_k^k \leq n!$, which as $k \geq \min\{k, d-l\}$ gives:

$$\begin{aligned} \beta_k &\leq \exp\left(\frac{1}{k}\ln(n!)\right) \\ &\leq \exp\left(\frac{1}{\min\{k, d-l\}}\ln(n!)\right) \\ &\leq \exp\left(\frac{n \ln n}{\min\{k, d-l\}}\right) \end{aligned}$$

Applying this to k and l along with the condition of $l - k \leq \frac{\min\{k, d-l\}}{n \ln n}$, then gives

$$a_l \leq \beta_k^{\frac{\min\{k, d-l\}}{n \ln n}} a_k \leq e \cdot a_k.$$

Similarly, we can show $a_l \geq \frac{1}{e} a_k$ using the condition $d - l \geq \min\{k, d-l\}$, which gives the other half of the claim.

Such a bound then allows us to establish our bound on expected depth of a key.

THEOREM 3.1. *For any n and t such that $0 < t < \binom{n}{2}$, and any key j, the expected depth of j in trees generated from permutations uniformly chosen with t inversions is at most*

$$\frac{4en^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}} + O(\log n).$$

Proof. Consider the first part of the terms that we are summing up in the expression for the expected depth of i in this tree from Equation 3.1 (which was in turn obtained via Corollary 3.1). We can expand out the coefficient of x^t in $f_{\textsc{Prod}(n)}(x)$ in terms of the various coefficients in $f_{\textsc{Prod}([n]\setminus i)}(x)$:

$$\frac{[x^t]\, f_{\textsc{Prod}([n]\setminus i)}(x)}{[x^t]\, f_{\textsc{Prod}([n])}(x)} = \frac{[x^t]\, f_{\textsc{Prod}([n]\setminus i)}(x)}{\sum_{k=0}^{i-1} [x^{t-k}]\, f_{\textsc{Prod}([n]\setminus i)}(x)}.$$

Lemma 2.3 gives that these coefficients for $f_{\textsc{Prod}([n]\setminus i)}(x)$ are log-concave, by which we can apply Lemma 3.1. However, Lemma 3.1 gives a bound on some terms of $\left[x^{t-k}\right] f_{\textsc{Prod}([n]\setminus i)}(x)$ only when $t - k$ is close to t, so we split the analysis into two cases:

1. $1 \leq i \leq \frac{\min\{t, \binom{n}{2} - t\}}{2n \ln n}$: In this case, we have $1 \leq i \leq \frac{\min\{t-i, \binom{n}{2} - i - t\}}{n \ln n}$, so $k \leq i \leq \frac{\min\{t-k, \binom{n}{2} - i - t\}}{n \ln n}$. Then Lemma 3.1 allows us to bound this denominator by

   $$\sum_{k=0}^{i-1} \left[x^{t-k}\right] f_{\textsc{Prod}([n]\setminus i)}(x) \geq \sum_{k=0}^{i-1} \frac{1}{e} \left[x^t\right] f_{\textsc{Prod}([n]\setminus i)}(x),$$

   which put back in gives:

   $$\frac{[x^t]\, f_{\textsc{Prod}([n]\setminus i)}(x)}{[x^t]\, f_{\textsc{Prod}([n])}(x)} \leq \frac{e}{i}.$$

   Summing across all i in this range then gives a bound of $O(\log n)$ for this term.

2. $\frac{\min\{t, \binom{n}{2} - t\}}{2n \ln n} < i \leq n$: In this case, we should omit some values of k that are too large. Then Lemma 3.1 allows us to bound this denominator by

   $$\begin{aligned} &\sum_{k=0}^{i-1} \left[x^{t-k}\right] f_{\textsc{Prod}([n]\setminus i)}(x) \\ \geq &\sum_{k=0}^{\frac{\min\left\{t, \binom{n}{2} - t\right\}}{2n \ln n}} \left[x^{t-k}\right] f_{\textsc{Prod}([n]\setminus i)}(x) \\ \geq &\sum_{k=0}^{\frac{\min\left\{t, \binom{n}{2} - t\right\}}{2n \ln n}} \frac{1}{e} \left[x^t\right] f_{\textsc{Prod}([n]\setminus i)}(x), \end{aligned}$$

   which put back in gives:

   $$\frac{[x^t]\, f_{\textsc{Prod}([n]\setminus i)}(x)}{[x^t]\, f_{\textsc{Prod}([n])}(x)} \leq \frac{2e \cdot n \ln n}{\min\left\{t, \binom{n}{2} - t\right\}}.$$

   Summing across all i in this range then gives a bound of $\frac{2en^2 \ln n}{\min\{t, \binom{n}{2} - t\}}$ for this term.

Since $\min\left\{t, \binom{n}{2} - t\right\} = O(n^2)$, the second term is the dominant term. Therefore, summing across all i then gives the overall bound.

The second part of the summation can be transformed via symmetry to become:

$$\sum_{i=1}^{n} \frac{\left[x^{\binom{n}{2} - t}\right] f_{\textsc{Prod}([n]\setminus i)}(x)}{[x^t]\, f_{\textsc{Prod}([n])}(x)} = \sum_{i=1}^{n} \frac{\left[x^{\binom{n}{2} - t}\right] f_{\textsc{Prod}([n]\setminus i)}(x)}{\left[x^{\binom{n}{2} - t}\right] f_{\textsc{Prod}([n])}(x)},$$

by which the exact same analysis above applies. The symmetry

$$\left[x^t\right] f_{\textsc{Prod}([n])}(x) = \left[x^{\binom{n}{2}-t}\right] f_{\textsc{Prod}([n])}(x)$$

used in the above equality is because the number of n-permutation with t inversion pairs is the same as the number of n-permutation with $\binom{n}{2}-t$ inversion pairs.

Theorem 3.1 also implies a bound on the expected interior path length (IPL), which is the sum of path lengths for the internal (non-leaf) nodes. For a binary tree, this value is within a constant factor of the sum of depths of all nodes. Therefore a consequence of Theorem 3.1 is that the expected IPL of trees generated from permutations uniformly chosen with t inversions is

$$O\left(\frac{n^3 \log n}{\min\left\{t, \binom{n}{2}-t\right\}}\right).$$

4 With High Probability Bounds

This section aims to propose a more involved method to show the high probability result. We will do so via the notion of negative correlation. Binary random variables $X_1, \ldots, X_n$ are *negatively correlated* if, for any subset $I \subset [n]$, we have

$$\Pr_{X_1 \ldots X_n}\left[\bigwedge_{i\in I} X_i = 1\right] \le \prod_{i\in I} \Pr_{X_i}\left[X_i = 1\right].$$

The concentration of negatively correlated variables is analogous to the concentration of independent random variables. We utilize the following version from [8]

Lemma 4.1. *(Theorem 3.4. of [8]) Let $X_1, X_2, \ldots X_n$ be* 0-1 *random variables with $X = \sum_i X_i$ such that for all $I \subseteq [n]$,*

$$Pr_{X_i : i\in I}\left[\bigwedge_{i\in I} X_i = 1\right] \le \lambda \prod_{i\in I} Pr_{X_i}\left[X_i = 1\right],$$

then for any ϵ we have:

$$Pr_{X_1 \ldots X_n}\left[X \ge (1+\epsilon)\,\mathbb{E}\left[X\right]\right] \le \lambda F\left(\mathbb{E}\left[X\right], \epsilon\right),$$

where

$$F\left(\mu, \epsilon\right) \stackrel{\text{def}}{=} \left[\frac{e^{\epsilon}}{\left(1+\epsilon\right)^{1+\epsilon}}\right]^{\mu}.$$

In our case, we define $X_{ij} = 1$ if i is an ancestor of j. We want to show that for each j, the set of variables X_{ij} (for all $i < j$) are negatively correlated. This is done by combining the use of log-concavity from Section 3 with the following identity regarding the probability for a set of keys I being ancestors for some j. We note that the starting point of Section 3, specifically Corollary 3.1, is also a direct consequence of the below claim.

Lemma 4.2. *For any n and any set of distinct keys $I = \{i_1, i_2 \ldots i_m\}$ with each $i_k < j$, consider the set*

$$I^{(j-)} \stackrel{\text{def}}{=} \left\{j - i + 1 : i \in I\right\}.$$

The number of permutations with t inversions such that all i_k are j's ancestors is equal to

$$\left[x^t\right] f_{\textsc{Prod}\left([n]\setminus I^{(j-)}\right)}(x).$$

Proof. Assume the number of permutations with k inversions, such that in the corresponding binary tree, all i_k are j's ancestors, is c_k.

For each k, consider a sub permutation in σ consisting of numbers in the range $[i_k, j]$. In this sub-permutation, the number i_k can be in position 1 through $j - i_k + 1$.

When number i_k is in position 1 in this sub permutation, i_k is an ancestor of j in the binary tree [5, Lemma 3.4]. Denote the actual position of i_k in this sub permutation by p_k. We can transform the permutation σ to a permutation $\widehat{\sigma}$ such that i_k appears earliest in $\widehat{\sigma}$ among all the numbers in the range $[i_k, j]$ by:

1. Pick up this sub permutation and denote it as $b_1, b_2, \ldots, b_{j-i_k+1}$, where $b_{p_k} = i_k$.
2. Move i_k to the front of b and preserve the order of the rest of the numbers to get $\hat{b}$.
3. Put this $\hat{b}$ back to σ in the corresponding indices to obtain $\widehat{\sigma}$.

We claim that this transformation decreases the number of inversions by $p_k - 1$, since the only pairs of numbers that can possibly change order is (i_k, x), where $x \in [i+1, j]$. Also, by doing this operation, p_l will not change for any $l \neq k$.

Therefore, we can perform such operation m times over the keys in I; after all operations are performed, the corresponding binary tree of the sequence satisfies that i_k is ancestor of j for any $1 \le k \le m$.

Also, by knowing $p_1, \ldots, p_k$, we can convert the resulting permutation back to the initial permutation by similar operations. Thus we can count the number of permutations with t inversions by enumerating all possible values of $p_1, \ldots, p_m$.

$$\left[x^t\right] f_{\textsc{Prod}([n])}(x) = \sum_{p_1=1}^{j-i_1+1} \cdots \sum_{p_m=1}^{j-i_m+1} c_{t-\sum_{l=1}^{m}(p_l-1)}.$$

When stated in generating functions terms, this means the generating function

$$f_C(x) \stackrel{\text{def}}{=} \sum_k c_k x^k$$

(where the definition of C and c implicitly included conditions on n, I, and j) satisfies:

$$f_{\textsc{Prod}([n])}(x) = f_C(x) \prod_{k=1}^{m} f_{\textsc{Sum}(j-i_k+1)}(x),$$

which gives the result by dividing through by the $f_{\textsc{Sum}(j-i_k+1)}(x)$ terms.

By lemma 4.2,

$$\Pr_{\sigma:inv(\sigma)=t}\left[\bigwedge_{i\in I} X_{ij}\right] = \frac{\left[x^k\right] f_{\textsc{Prod}\left([n]\setminus I^{(j-)}\right)}(x)}{\left[x^k\right] f_{\textsc{Prod}([n])}(x)},$$

and

$$\Pr_{\sigma:inv(\sigma)=t}\left[X_{ij}\right] = \frac{\left[x^k\right] f_{\textsc{Prod}([n]\setminus(j-i+1))}(x)}{\left[x^k\right] f_{\textsc{Prod}([n])}(x)},$$

The condition that X_{ij} are negatively correlated directly translates to showing that for any set I,

$$\frac{\left[x^t\right] f_{\textsc{Prod}\left([n]\setminus I^{(j-)}\right)}(x)}{\left[x^t\right] f_{\textsc{Prod}([n])}(x)} \leq \prod_{i\in I} \frac{\left[x^k\right] f_{\textsc{Prod}([n]\setminus(j-i+1))}(x)}{\left[x^k\right] f_{\textsc{Prod}([n])}(x)}$$

which is equivalent (after removing the common denominator) to

$$\begin{aligned}&\left[x^k\right] f_{\textsc{Prod}\left([n]\setminus I^{(j-)}\right)}(x) \cdot \left(\left[x^k\right] f_{\textsc{Prod}([n])}(x)\right)^{|I|-1}\\ &\leq \prod_{i\in I}\left[x^k\right] f_{\textsc{Prod}([n]\setminus(j-i+1))}(x)\end{aligned}$$

We need a lemma to proceed.

Lemma 4.3. *For any set of indices* $K = \{k_1, k_2, \ldots, k_m\} \subseteq [n]$*, and for any* t*, we have*

$$\begin{aligned}&\left[x^t\right] f_{\textsc{Prod}([n]\setminus K)}(x) \cdot \left[x^t\right] f_{\textsc{Prod}([n])}(x)\\ &\leq \left[x^t\right] f_{\textsc{Prod}([n]\setminus\{k_1,k_2\ldots k_{m-1}\})}(x) \left[x^t\right] f_{\textsc{Prod}([n]\setminus k_m)}(x)\end{aligned}$$

Proof. Let

$$b_t \overset{\text{def}}{=} \left[x^t\right] f_{\textsc{Prod}([n]\setminus K)}(x),$$

and expanding the above formula w.r.t. this sequence of bs becomes:

$$\begin{aligned}&b_t\left(\sum_{l_1=0}^{k_1-1}\cdots\sum_{l_m=0}^{k_m-1} b_{t-l_1-\ldots-l_m}\right)\\ &\leq \left(\sum_{l_1=0}^{k_1-1}\cdots\sum_{l_{m-1}=0}^{k_{m-1}-1} b_{t-l_1-\ldots-l_{m-1}}\right)\left(\sum_{l_m=0}^{k_m-1} b_{t-l_m}\right),\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\sum_{l_1=0}^{k_1-1}\cdots\sum_{l_m=0}^{k_m-1} b_t b_{t-l_1-\ldots-l_m}\\ &\leq \sum_{l_1=0}^{k_1-1}\cdots\sum_{l_m=0}^{k_m-1} b_{t-l_m} b_{t-l_1-\ldots-l_{m-1}}.\end{aligned}$$

By Lemma 2.3, b is a log-concave sequence. So applying Lemma 2.2 shows that the above holds entry-wise.

Applying Lemma 4.3 inductively then gives the negative correlation statement.

Theorem 4.1. *For any* n *and* t *such that* $0 < t < \binom{n}{2}$*, the height of trees generated from permutations uniformly chosen with* t *inversions is*

$$O\left(\frac{n^2\log n}{\min\left\{t, \binom{n}{2} - t\right\}}\right).$$

with high probability. Specifically, for any positive number k*, there exists a constant* d *such that the height of the tree is greater than*

$$\frac{dn^2\log n}{\min\left\{t, \binom{n}{2} - t\right\}}$$

with probability at most $\frac{2}{n^k}$*.*

Proof. For any key j, from Theorem 3.1, we know that

$$\mathbb{E}_{\sigma:inv(\sigma)=t}\left[\sum_{i<j} X_{ij}\right] < \frac{cn^2\log n}{\min\left\{t, \binom{n}{2} - t\right\}}.$$

for some constant c. Lemma 4.1 basically lets us use Chernoff bound for this summation. If we set $\mu = \mathbb{E}_{\sigma:inv(\sigma)=t}\left[\sum_{i<j} X_{ij}\right]$, we can use the following looser Chernoff bound for some $\delta > 1$:

$$\Pr_{\sigma:inv(\sigma)=t}\left[\sum_{i<j} X_{ij} \geq (1+\delta)\mu\right] \leq e^{-\delta\mu}.$$

Specifically, letting $\delta = 1 + \frac{(k+1)\ln n}{\mu}$ gives

$$\begin{aligned}&\Pr_{\sigma:inv(\sigma)=t}\left[\sum_{i<j} X_{ij} \geq 2\mu + (k+1)\ln n\right]\\ &\leq e^{-\mu-(k+1)\ln n}\\ &\leq \frac{1}{n^{k+1}}.\end{aligned}$$

Since $\mu < \frac{cn^2 \log n}{\min\{t, \binom{n}{2} - t\}}$, and $\ln n \leq \frac{n^2 \log n}{\min\{t, \binom{n}{2} - t\}}$, we can use $\frac{(2c+k+1)n^2 \log n}{\min\{t, \binom{n}{2} - t\}}$ as an upper bound for $2\mu + (k+1)\ln n$. This then gives the bound over all $i < j$:

$$\mathrm{Pr}_{\sigma:inv(\sigma)=t}\left[\sum_{i<j} X_{ij} \geq \frac{(2c+k+1)n^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}}\right] \leq \frac{1}{n^{k+1}}.$$

It remains to handle the larger labels $i > j$. We can use the same trick as before: if we convert any number δ_t in the permutation to $n - \delta_t + 1$, then all the larger keys will become smaller keys, and thus we can apply the above inequality again to get

$$\begin{aligned}
&\mathrm{Pr}_{\sigma:inv(\sigma)=t}\left[\sum_{i>j} X_{ij} \geq \frac{(2c+k+1)n^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}}\right] \\
&= \mathrm{Pr}_{\sigma:inv(\sigma)=\binom{n}{2}-t}\left[\sum_{n-i+1<n-j+1} X_{n-i+1,n-j+1}\right. \\
&\qquad \left. \geq \frac{(2c+k+1)n^2 \log n}{\min\left\{\binom{n}{2} - t, t\right\}}\right] \\
&\leq \frac{1}{n^{k+1}}.
\end{aligned}$$

Therefore, by union bound over these two cases, we know that the depth of j is greater than $\frac{(4c+2k+2)n^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}}$ with probability at most $\frac{2}{n^{k+1}}$ The result then follows from taking a union bound over all keys j with the constant d equal to $4c + 2k + 2$.

This theorem also implies that the expected height of the tree is $O\left(\frac{n^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}}\right)$: Even though it is possible for the tree height to exceed $\frac{(4c+2k+2)n^2 \log n}{\min\left\{t, \binom{n}{2} - t\right\}}$, it happens with probability at most $\frac{2}{n^k}$. When it does happen, the tree height still has a natural upper bound of n. Therefore, this rare case does not affect the expected height of the tree asymptotically.

References

[1] J. Culberson and J. I. Munro, *Analysis of the standard deletion algorithms in exact fit domain binary search trees*, Algorithmica, 5 (1990), pp. 295–311.

[2] A. Elmasry, *Adaptive sorting with AVL trees*, in Proc. IFIP TCS, 2004, pp. 307–316.

[3] A. Elmasry and A. Hammad, *An empirical study for inversions-sensitive sorting algorithms*, in Proc. WEA, 2005, pp. 597–601.

[4] M. T. Goodrich and R. Tamassia, *Algorithm Design and Applications*, Wiley Publishing, 1st ed., 2014.

[5] M. T. Goodrich, R. Tamassia, and D. M. Mount, *Data structures and algorithms in C++*, Wiley, 2004.

[6] M. He and M. Li, *Deletion without rebalancing in non-blocking binary search trees*, in Proc. PODC, 2016, pp. 34:1–34:17.

[7] J. Iacono, *Key-independent optimality*, Algorithmica, 42 (2005), pp. 3–10.

[8] A. Panconesi and A. Srinivasan, *Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds*, SIAM J. Comput., 26 (1997), pp. 350–368.

[9] B. Pfaff, *Performance analysis of BSTs in system software*, in Proc. SIGMETRICS, 2004, pp. 410–411. full version available at `https://benpfaff.org/papers/libavl.pdf`.

[10] R. Saikkonen and E. Soisalon-Soininen, *Bulk-insertion sort: Towards composite measures of presortedness*, in Proc. SEA, 2009, pp. 269–280.

[11] R. Seidel and C. R. Aragon, *Randomized search trees*, Algorithmica, 16 (1996), pp. 464–497. Available at:https://faculty.washington.edu/aragon/pubs/rst96.pdf.

[12] D. D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Commun. ACM, 28 (1985), pp. 202–208.

[13] R. P. Stanley, *Log-concave and unimodal sequences in algebra, combinatorics, and geometry*, Annals of the New York Academy of Sciences, 576 (1989), pp. 500–535. Available at: `http://dedekind.mit.edu/~rstan/pubs/pubfiles/72.pdf`.

A Deferred Proofs

In this section we provide some of the proofs about properties of generating functions for completeness.

Proof. (of Lemma 2.2) The log-concavity defines a total order of terms in the form of $\frac{a_k}{a_{k-1}}$. The following inequality is an immediate result from this total order.

$$\frac{a_{k_2}}{a_{k_2-1}} \geq \frac{a_{k_3+1}}{a_{k_3}}$$

which proves the lemma when $k_1 = k_2 - 1$ and $k_3 = k_4 - 1$. Then induction on $k_2 - k_1$ can finish the proof.

Proof. (of Lemma 2.3 We will show this using induction on the size of I, $|I|$. The base case of $|I| = 1$ follows from all the coefficients being 1.

For the inductive case, let i be any element in I, and let the coefficients of $f_{\text{Prod}(I\setminus i)}(x)$ be $a_0 \ldots a_{d-i}$. Then by the inductive hypothesis, we have for any $0 < k < d - i$, $a_k^2 \geq a_{k-1}a_{k+1}$. We can also overload notations so that for any $k < 0$ or $k > d$, $a_k = 0$, while preserving this condition.

Now consider the function $f_{\text{Prod}(I)}(x) = f_{\text{Prod}(I\setminus i)}(x) \cdot f_{\text{Sum}(i)}(x)$, its coefficients can be characterized as:

$$\left[x^k\right] f_{\textsc{Prod}(I)}\left(x\right) = \sum_{k-i+1 \le j \le k} a_j,$$

which when plugged into the desired condition, and expanding, gives:

$$\begin{aligned}&\left(\left[x^k\right] f_{\textsc{Prod}(I)}\left(x\right)\right)^2 - \left[x^{k+1}\right] f_{\textsc{Prod}(I)}\left(x\right) \cdot \left[x^{k-1}\right] f_{\textsc{Prod}(I)}\left(x\right)\\ &= \sum_{k-i+1 \le j_1, j_2 \le k} a_{j_1} a_{j_2} - \sum_{k-i \le j_1 \le k-1, k-i+2 \le j_2 \le k+1} a_{j_1} a_{j_2}\\ &\ge 0.\end{aligned}$$

In order to simplify this, observe that any term $a_{j_1} a_{j_2}$ with $k-i+1 \le j_1 \le k-1$ and $k-i+2 \le j_2 \le k$ appears in both terms. The remaining terms then becomes:

$$\begin{aligned}&\sum_{k-i+2 \le j_2 \le k} a_k a_{j_2} + \sum_{k-i+1 \le j_1 \le k} a_{j_1} a_{k-i+1}\\ &- \sum_{k-i+2 \le j_2 \le k+1} a_{k-i} a_{j_2} - \sum_{k-i+1 \le j_1 \le k-1} a_{j_1} a_{k+1}.\end{aligned}$$

The second the third summations can be collected into:

$$\sum_{0 \le j \le i-1} a_{k-j} a_{k-i+1} - a_{k-i} a_{k+1-j},$$

and here we have $k-i \le k-i+1 \le k-j \le k+1-j$. So we can invoke Lemma 2.2 to show that this term is non-negative.

The first the fourth terms can also be combined similarly, which then gives the overall bound.