Smoothed Analysis of Binary Search Trees and Quicksort Under Additive Noise

Bodo Manthey[1]* and Till Tantau[2]

[1] Saarland University, Computer Science
Postfach 151150, 66041 Saarbrücken, Germany
manthey@cs.uni-sb.de
[2] Universität zu Lübeck, Institut für Theoretische Informatik
Ratzeburger Allee 160, 23538 Lübeck, Germany
tantau@tcs.uni-luebeck.de

Abstract. Binary search trees are a fundamental data structure and their height plays a key role in the analysis of divide-and-conquer algorithms like quicksort. We analyze their smoothed height under additive uniform noise: An adversary chooses a sequence of n real numbers in the range $[0, 1]$, each number is individually perturbed by adding a value drawn uniformly at random from an interval of size d, and the resulting numbers are inserted into a search tree. An analysis of the smoothed tree height subject to n and d lies at the heart of our paper: We prove that the smoothed height of binary search trees is $\Theta(\sqrt{n/d}+\log n)$, where $d \geq 1/n$ may depend on n. Our analysis starts with the simpler problem of determining the smoothed number of left-to-right maxima in a sequence. We establish matching bounds, namely once more $\Theta(\sqrt{n/d} + \log n)$. We also apply our findings to the performance of the quicksort algorithm and prove that the smoothed number of comparisons made by quicksort is $\Theta(\frac{n}{d+1}\sqrt{n/d} + n \log n)$.

1 Introduction

To explain the discrepancy between average-case and worst-case behavior of the simplex algorithm, Spielman and Teng introduced the notion of *smoothed analysis* [14]. Smoothed analysis interpolates between average-case and worst-case analysis: Instead of taking a worst-case instance or, as in average-case analysis, choosing an instance completely at random, we analyze the complexity of (possibly worst-case) objects subject to slight random perturbations. On the one hand, perturbations model that nature is not (or not always) adversarial. On the other hand, perturbations reflect the fact that data is often subject to measurement or rounding errors; even if the instance at hand was initially a worst-case instance, due to such errors we would probably get a less difficult instance in practice. Spielman and Teng [15] give a comprehensive survey on results and open problems in smoothed analysis.

* Work done at the Department of Computer Science at Yale University, supported by the Postdoc-Program of the German Academic Exchange Service (DAAD).

Binary search trees are one of the most fundamental data structures in computer science and they are the building blocks for a large variety of data structures. One of the most important parameter of binary search trees is their *height*. The worst-case height of a binary tree for n numbers is n. The average-case behavior has been the subject of a considerable amount of research, culminating in the result that the average-case height is $\alpha \ln n - \beta \ln \ln n + O(1)$, where $\alpha \approx 4.311$ is the larger root of $\alpha \ln(2e/\alpha) = 1$ and $\beta = 3/(2\ln(\alpha/2)) \approx 1.953$ [12]. Furthermore, the variance of the height is bounded by a constant, as was proved independently by Drmota [6] and Reed [12], and also all higher moments are bounded by constants [6]. Drmota [7] gives a recent survey.

Beyond being an important data structure, binary search trees play a central role in the analysis of divide-and-conquer algorithms like quicksort [9, Section 5.2.2]. While quicksort needs $\Theta(n^2)$ comparisons in the worst case, the average number of comparisons is $2n \log n - \Theta(n)$ with a variance of $(7 - \frac{2}{3}\pi^2) \cdot n^2 - 2n \log n + O(n)$ as mentioned by Fill and Janson [8]. Quicksort and binary search trees are closely related: The height of the tree $T(\sigma)$ obtained from a sequence σ is equal to the number of levels of recursion required by quicksort to sort σ. The number of comparisons, which corresponds to the total path length of $T(\sigma)$, is at most n times the height of $T(\sigma)$.

Binary search trees are also related to the number of left-to-right maxima of a sequence, which is the number of new maxima seen while scanning a sequence from left to right. The number of left-to-right maxima of σ is equal to the length of the rightmost path of the tree $T(\sigma)$, which means that left-to-right maxima provide an easy-to-analyze lower bound for the height of binary search trees. In the worst-case, the number of left-to-right maxima is n, while it is $\sum_{i=1}^{n} 1/i \in \Theta(\log n)$ on average. The study of left-to-right maxima is also of independent interest. For instance, the number of times a data structure for keeping track of a bounding box of moving object needs to be updated is closely related to the number of left-to-right maxima (and minima) of the coordinate components of the objects. (See Basch et al. [2] for an introduction to data structures for mobile data.) Left-to-right maxima also play a role in the analysis of quicksort [13].

Given the discrepancies between average-case and worst-case behavior of binary search trees, quicksort, and the number of left-to-right maxima, the question arises of what happens in between when the randomness is limited.

Our results. We continue the smoothed analysis of binary search trees and quicksort begun by Banderier et al. [1] and Manthey and Reischuk [10]. However, we return to the original idea of smoothed analysis that input numbers are perturbed by adding random numbers. The perturbation model introduced by Spielman and Teng for the smoothed analysis of continuous problems like linear programming is appropriate for algorithms that process real numbers. In their model, each of the real numbers in the adversarial input is perturbed by adding a small Gaussian noise. This model of perturbation favors instances in the neighborhood of the adversarial input for a fairly natural and realistic notion of "neighborhood."

In our model the adversarial input sequence consists of n real numbers in the interval $[0, 1]$. Then, each of the real numbers is individually perturbed by adding a random number drawn uniformly from an interval of size d, where $d = d(n)$ may depend on n. If $d < 1/n$, then the sorted sequence $(1/n, 2/n, 3/n, \ldots, n/n)$ stays a sorted sequence and the smoothed height of binary search trees (as well as the performance of quicksort and the number of left-to-right maxima) is the same as in the worst-case. We always assume $d \geq 1/n$ in the following.

We study the smoothed height of binary search trees, the smoothed number of comparisons made by quicksort, and the smoothed number of left-to-right maxima under additive noise. In each case we prove tight upper and lower bounds:

1. The smoothed number of left-to-right maxima is $\Theta(\sqrt{n/d} + \log n)$ as shown in Section 3. This result will be exploited in the subsequent sections.
2. The smoothed height of binary search trees is $\Theta(\sqrt{n/d} + \log n)$ as shown in Section 4.
3. The smoothed number of comparisons made by quicksort is $\Theta(\frac{n}{d+1}\sqrt{n/d} + n \log n)$ as shown in Section 5. Thus, the perturbation effect of $d \in \omega(1)$ is stronger than for $d \in o(1)$.

Already for $d \in \omega(1/n)$, we obtain bounds that are asymptotically better than the worst-case bounds. For constant values of d, which correspond to a perturbation by a constant percentage like 1%, the height of binary search trees drops from the worst-case height of n to $O(\sqrt{n})$, and quicksort needs only $O(n^{3/2})$ comparisons.

It is tempting to assume that results such as the above will hold in the same way for other distributions, such as the Gaussian distribution, with d replaced by the standard deviation. We contribute a surprising result in Section 6: We present a well-behaved probability distribution (symmetric, monotone on the positive reals, smooth) for which sorting sequences can *decrease* the expected number of left-to-right maxima. This effect is quite counter-intuitive and the literature contains the claim that one can restrict attention to sorted sequences since they are the worst-case sequences also in the smoothed setting [4, 3]. Our distribution refutes this claim.

Related work. The first smoothed analysis of quicksort, due to Banderier, Beier, and Mehlhorn [1], uses a perturbation model different from the one used in the present paper, namely a *discrete perturbation model.* Such models take discrete objects like permutations as input and again yield discrete objects like another permutation. Banderier et al. used *p-partial permutations*, which work as follows: An adversary chooses a permutation of the numbers $\{1, \ldots, n\}$ as sequence, every element of the sequence is marked independently with a probability of p, and then the marked elements are randomly permuted. Banderier et al. showed that the number of comparisons subject to p-partial permutations is $O(\frac{n}{p} \cdot \log n)$. Furthermore, they proved bounds on the smoothed number of left-to-right maxima subject to this model.

Manthey and Reischuk [10] analyzed the height of binary search trees under p-partial permutations. They proved a lower bound of $0.8 \cdot (1 - p) \cdot \sqrt{n/p}$ and an asymptotically matching upper bound of $6.7 \cdot (1-p) \cdot \sqrt{n/p}$ for the smoothed tree height. For the number of left-to-right maxima, they showed a lower bound of $0.6 \cdot (1-p) \cdot \sqrt{n/p}$ and an upper bound of $3.6 \cdot (1-p) \cdot \sqrt{n/p}$.

Special care must be taken when defining perturbation models for discrete inputs: The perturbation should favor instances in the neighborhood of the adversarial instance, which requires a suitable definition of neighborhood in the first place, and the perturbation should preserve the global structure of the adversarial instance. Partial permutations have the first feature [10, Lemma 3.2], but destroy much of the global order of the adversarial sequence.

The smoothed number of left-to-right maxima for the additive noise model of the present paper was already considered by Damerow et al. [4, 5, 3]. They considered so-called kinetic data structures, which keep track of properties of a set of moving points like a bounding box for them or a convex hull, and they introduced the notion of *smoothed motion complexity*. They also considered left-to-right maxima since left-to-right maxima provide upper and lower bounds on the number of times a bounding box needs to be updated for moving points. For left-to-right maxima, Damerow et al. show an upper bound of $O(\sqrt{n \log n/d} + \log n)$ and a lower bound of $\Omega(\sqrt{n/d})$. In the present paper, we show that the exact bound is $\Theta(\sqrt{n/d} + \log n)$.

2 Preliminaries

Intervals are denoted by $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$. To denote an interval that does not include an endpoint, we replace the square bracket next to the endpoint by a parenthesis. We denote *sequences* of real numbers by $\sigma = (\sigma_1, \ldots, \sigma_n)$, where $\sigma_i \in \mathbb{R}$. For $U = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ with $i_1 < i_2 < \cdots < i_\ell$ let $\sigma_U = (\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_\ell})$ denote the *subsequence* of σ of the elements at positions in U. We denote probabilities by $\mathbb{P}$ and expected values by $\mathbb{E}$.

Throughout the paper, we will assume for the sake of clarity that numbers like $\sqrt{d}$ are integers and we do not write down the tedious floor and ceiling functions that are actually necessary. Since we are interested in asymptotic bounds, this does not affect the validity of the proofs.

Due to lack of space, some proofs are omitted. For complete proofs, we refer to the full version of this paper [11].

2.1 Binary Search Trees, Left-To-Right Maxima, and Quicksort

Let σ be a sequence of length n consisting of pairwise distinct elements. For the following definitions, let $G = \{i \in \{1, \ldots, n\} \mid \sigma_i > \sigma_1\}$ be the set of positions of elements greater than σ_1, and let $S = \{i \in \{1, \ldots, n\} \mid \sigma_i < \sigma_1\}$ be the set of positions of elements smaller than σ_1.

From σ, we obtain a *binary search tree* $T(\sigma)$ by iteratively inserting the elements $\sigma_1, \ldots, \sigma_n$ into the initially empty tree as follows: The root of $T(\sigma)$ is σ_1. The left subtree of the root σ_1 is $T(\sigma_S)$, and the right subtree of σ_1 is $T(\sigma_G)$. The *height of* $T(\sigma)$ is the maximum number of nodes on any root-to-leaf path of

$T(\sigma)$: Let $\text{height}(\sigma) = 1 + \max\{\text{height}(\sigma_S), \text{height}(\sigma_G)\}$, and let $\text{height}(\sigma) = 0$ when σ is the empty sequence.

The number of *left-to-right maxima* of σ is the number of maxima seen when scanning σ from left to right: let $\text{ltrm}(\sigma) = 1 + \text{ltrm}(\sigma_G)$, and let $\text{ltrm}(\sigma) = 0$ when σ is the empty sequence. The number of left-to-right maxima of σ is equal to the length of the rightmost path of $T(\sigma)$, so $\text{ltrm}(\sigma) \leq \text{height}(\sigma)$.

Quicksort is the following sorting algorithm: Given σ, we construct σ_S and σ_G. To do this, all elements of $(\sigma_2, \ldots, \sigma_n)$ have to be compared to σ_1, which is called the *pivot element*. Then we sort σ_S and σ_G recursively to obtain τ_S and τ_G, respectively. Finally, we output $\tau = (\tau_S, \sigma_1, \tau_G)$. The number $\text{qs}(\sigma)$ of comparisons needed to sort σ is thus $\text{qs}(\sigma) = (n-1) + \text{qs}(\sigma_S) + \text{qs}(\sigma_G)$ if σ has a length of $n \geq 1$, and $\text{qs}(\sigma) = 0$ when σ is the empty sequence.

2.2 Perturbation Model

The perturbation model of *additive noise* is defined as follows: Let $d = d(n) \geq 0$ be the perturbation parameter (d may depend on n). Given a sequence σ of n numbers chosen by an adversary from the interval $[0,1]$, we draw a *noise* ν_i for each $i \in \{1, \ldots, n\}$ uniformly and independently from each other at random from the interval $[0,d]$. Then we obtain the perturbed sequence $\overline{\sigma} = (\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$ by adding ν_i to σ_i, that is, $\overline{\sigma}_i = \sigma_i + \nu_i$. Note that $\overline{\sigma}_i$ need no longer be an element of $[0,1]$, but $\overline{\sigma}_i \in [0, d+1]$. For $d > 0$ all elements of $\overline{\sigma}$ are distinct with a probability of 1.

For this model, we define the random variables $\text{height}_d(\sigma)$, $\text{qs}_d(\sigma)$, as well as $\text{ltrm}_d(\sigma)$, which denote the smoothed search tree height, smoothed number of quicksort comparisons, and smoothed number of left-to-right maxima, respectively, when the sequence σ is perturbed by d-noise. Since the adversary chooses σ, our goal are bounds for $\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{height}_d(\sigma)\big)$, $\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{qs}_d(\sigma)\big)$, and $\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{ltrm}_d(\sigma)\big)$. In the following, we will sometimes write $\text{height}(\overline{\sigma})$ instead of $\text{height}_d(\sigma)$ if d is clear from the context. Since $\overline{\sigma}$ is random, $\text{height}(\overline{\sigma})$ is also a random variable. Similarly, we will use $\text{ltrm}(\overline{\sigma})$ and $\text{qs}(\overline{\sigma})$.

The choice of the interval sizes is arbitrary since the model is invariant under scaling if we scale the perturbation parameter accordingly. This is summarized in the following lemma, which we will exploit a couple of times in the following.

Lemma 1. *Let $b > a$ and $d > 0$ be arbitrary real numbers, and let $d' = d/(b-a)$. Then $\max_{\sigma \in [a,b]^n} \mathbb{E}\big(\text{height}_d(\sigma)\big) = \max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{height}_{d'}(\sigma)\big)$. For quicksort and the number of left-to-right maxima, we have analogous equalities.*

As argued earlier, if $d < 1/n$, the adversary can specify $\sigma = (1/n, \ldots, n/n)$ and adding the noise terms does not affect the order of the elements. This means that we get the worst-case height, number of comparisons, and number of left-to-right maxima. Because of this observation we will restrict our attention to $d \geq 1/n$.

If d is large, the noise will swamp out the original instance, and the order of the elements of $\overline{\sigma}$ will depend only on the noise rather than the original instance. For intermediate d, additive noise interpolates between average and worst case.

3 Smoothed Number of Left-To-Right Maxima

We start our analyses with the smoothed number of left-to-right maxima, which provides us with a lower bound on the height of binary search trees as well. Our aim for the present section is to prove the following theorem.

Theorem 1. *For $d \geq 1/n$, we have*

$$\max_{\sigma \in [0,1]^n} \mathbb{E}\bigl(\mathrm{ltrm}_d(\sigma)\bigr) \in \Theta\bigl(\sqrt{n/d} + \log n\bigr).$$

The lower bound of $\Omega\bigl(\sqrt{n/d} + \log n\bigr)$ is already stated without proof in [4] and a proof can be found in [3], so we prove only the upper bound. The following notations will be helpful: For $j \leq 0$, let $\sigma_j = \nu_j = 0$. This allows us to define $\delta_i = \sigma_i - \sigma_{i-\sqrt{nd}}$ for all $i \in \{1, \ldots, n\}$. We define $I_i = \{j \in \{1, \ldots, n\} \mid i - \sqrt{nd} \leq j < i\}$ to be the set of the $|I_i| = \min\{i-1, \sqrt{nd}\}$ positions that precede i.

To prove the upper bound for the smoothed number of left-to-right maxima, we proceed in two steps: First, a "bubble-sorting argument" is used to show that the adversary should choose a sorted sequence. Note that this is not as obvious as it may seem since in Section 6 we show that this bubble-sorting argument does not apply to all distributions. Second, we prove that the expected number of left-to-right maxima of sorted sequences is $O(\sqrt{n/d} + \log n)$, which improves the bound of $O(\sqrt{n \log n/d} + \log n)$ [3, 4].

Lemma 2. *For every σ and its sorted version τ, $\mathbb{E}\bigl(\mathrm{ltrm}_d(\sigma)\bigr) \leq \mathbb{E}\bigl(\mathrm{ltrm}_d(\tau)\bigr)$.*

Lemma 3. *For all σ of length n and all $d \geq 1/n$, we have $\mathbb{E}\bigl(\mathrm{ltrm}_d(\sigma)\bigr) \in O\bigl(\sqrt{n/d} + \log n\bigr)$.*

Proof. By Lemma 2 we can restrict ourselves to proving the lemma for sorted sequences σ. We estimate the probability that a given $\overline{\sigma}_i$ for $i \in \{1, \ldots, n\}$ is a left-to-right maximum. Then the bound follows by the linearity of expectation. To bound the probability that $\overline{\sigma}_i$ is a left-to-right maximum (ltrm), consider the following computation:

$$\mathbb{P}\bigl(\overline{\sigma}_i \text{ is an ltrm}\bigr) \leq \mathbb{P}\bigl(\forall j \in I_i \colon \nu_j < \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}\bigr) \tag{1}$$

$$\leq \mathbb{P}\bigl(d < \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}\bigr) + \int_0^{d-\delta_i} \mathbb{P}\bigl(\forall j \in I_i \colon \nu_j < \sigma_i + x - \sigma_{i-\sqrt{nd}}\bigr) \cdot \tfrac{1}{d}\,\mathrm{d}x \tag{2}$$

$$\leq \tfrac{\delta_i}{d} + \int_0^{d} \mathbb{P}\bigl(\forall j \in I_i \colon \nu_j < x\bigr) \cdot \tfrac{1}{d}\,\mathrm{d}x \tag{3}$$

$$\leq \tfrac{\delta_i}{d} + \mathbb{P}\bigl(\forall j \in I_i \colon \nu_j < \nu_i\bigr) \;=\; \tfrac{\delta_i}{d} + \tfrac{1}{|I_i|+1}. \tag{4}$$

To see that (1) holds, assume that $\overline{\sigma}_i$ is a left-to-right maximum. Then $\overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$ must be larger than the noises of all the elements in the index range I_i, for if the noise ν_j of some element σ_j were larger than $\overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$, then $\overline{\sigma}_j = \sigma_j + \nu_j$ would be larger than $\sigma_j + \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$. Since the sequence is sorted, we would get $\sigma_j + \overline{\sigma}_i - \sigma_{i-\sqrt{nd}} \geq \overline{\sigma}_i$, and $\overline{\sigma}_i$ would not be a left-to-right maximum.

For (2), first observe that $\nu_j < \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$ is surely the case for all $j \in I_i$ if $d < \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$. So, consider the case $d \geq \overline{\sigma}_i - \sigma_{i-\sqrt{nd}} = \delta_i + \nu_i$. Then $\nu_i \in [0, d - \delta_i]$ and we can rewrite $\mathbb{P}(\forall j \in I_i \colon \nu_j < \delta_i + \nu_i)$ as $\int_0^{d-\delta_i} \mathbb{P}(\forall j \in I_i \colon \nu_j < \delta_i + x) \cdot \frac{1}{d} \, \mathrm{d}x$, where $1/d$ is the density of ν_i. For (3) observe that $d < \overline{\sigma}_i - \sigma_{i-\sqrt{nd}}$ is equivalent to $d - \delta_i < \nu_i$ and the probability of this is δ_i/d. Furthermore, we performed an index shift in the integral. In (4), we replaced the integral by a probability once more and get the final result.

We have $\sum_{i=1}^n \delta_i = \sum_{i=1}^n (\sigma_i - \sigma_{i-\sqrt{nd}}) = \sum_{i=n-\sqrt{nd}+1}^n \sigma_i \leq \sqrt{nd}$. The second equality holds since most σ_i cancel themselves out and $\sigma_i = 0$ for $i \leq 0$. The inequality holds since there are $\sqrt{nd}$ summands. We bound $1/(|I_i| + 1) = 1/\min\{i, \sqrt{nd}+1\}$ by $1/i + 1/\sqrt{nd}$ and sum over all i: $\mathbb{E}\big(\mathrm{ltrm}_d(\sigma)\big) \leq \sum_{i=1}^n \big(\frac{\delta_i}{d} + \frac{1}{|I_i|+1}\big) \leq \frac{\sqrt{nd}}{d} + \sum_{i=1}^n \frac{1}{i} + \frac{n}{\sqrt{nd}} \in O(\sqrt{n/d} + \log n)$. □

4 Smoothed Height of Binary Search Trees

In this section we prove our first main result, an exact bound on the smoothed height of binary search trees under additive noise. The bound is the same as for left-to-right maxima, as stated in the following theorem.

Theorem 2. *For $d \geq 1/n$, we have*

$$\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \in \Theta\big(\sqrt{n/d} + \log n\big).$$

In the rest of this section, we prove this theorem. We have to prove an upper and a lower bound, but the lower bound follows directly from the lower bound of $\Omega(\sqrt{n/d} + \log n)$ for the smoothed number of left-to-right maxima (the number of left-to-right maxima in a sequence is the length of the rightmost path of the sequence's search tree). Thus, we only need to focus on the upper bound. To prove the upper bound of $O(\sqrt{n/d} + \log n)$ on the smoothed height of binary search trees, we need some preparations. In the next subsection we introduce the concept of *increasing and decreasing runs* and show how they are related to binary search tree height. As we will see, bounding the length of these runs implicitly bounds the height of binary search trees. This allows us to prove the upper bound on the smoothed height of binary search trees in the main part of this section.

4.1 Increasing and Decreasing Runs

In order to analyze the smoothed height of binary search trees, we introduce a related measure for which an upper bound is easier to obtain. Given a sequence σ, consider a root-to-leaf path of the tree $T(\sigma)$. We extract two subsequences $\alpha = (\alpha_1, \ldots, \alpha_k)$ and $\beta = (\beta_1, \ldots, \beta_\ell)$ from this path according to the following algorithm: We start at the root. When we are at an element σ_i of the path, we look at the direction in which the path continues from σ_i. If it continues with the right child of σ_i, we append σ_i to α; if it continues with the left child, we

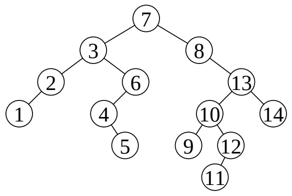

Fig. 1. The tree $T(\sigma)$ obtained from the sequence $\sigma = (7, 8, 13, 3, 2, 10, 9, 6, 4, 12, 14, 1, 5, 11)$. We have $\text{height}(\sigma) = 6$. The root-to-leaf path ending at 11 yields the increasing run $\alpha = (7, 8, 10, 11)$ and the decreasing run $\beta = (13, 12, 11)$.

append σ_i to β; and if σ_i is a leaf (has no children), then we append σ_i to both α and β. This construction ensures $\alpha_1 < \cdots < \alpha_k = \beta_\ell < \cdots < \beta_1$ and the length of σ is $k + \ell - 1$. Figure 1 shows an example of how α and β are constructed.

A crucial property of the sequence α is the following: Let $\alpha_i = \sigma_{j_i}$ for all $i \in \{1, \dots, k\}$ with $j_1 < j_2 < \cdots < j_k$. Then none of $\sigma_1, \dots, \sigma_{j_i - 1}$ lies in the interval (α_i, α_{i+1}), for otherwise α_i and α_{i+1} cannot be on the same root-to-leaf path. A similar property holds for the sequence β: No element of σ prior to β_i lies in the interval (β_{i+1}, β_i). We introduce a special name for sequences with this property.

Definition 1. *An* increasing run *of a sequence σ is a subsequence $(\sigma_{i_1}, \dots, \sigma_{i_k})$ with the following property: For every $j \in \{1, \dots, k-1\}$, no element of σ prior to σ_{i_j} lies in the interval $(\sigma_{i_j}, \sigma_{i_{j+1}})$. Analogously, a* decreasing run *of σ is a subsequence $(\sigma_{i_1}, \dots, \sigma_{i_\ell})$ with $\sigma_{i_1} > \cdots > \sigma_{i_\ell}$ such no element prior to σ_{i_j} lies in the interval $(\sigma_{i_{j+1}}, \sigma_{i_j})$.*

Let $\text{inc}(\sigma)$ and $\text{dec}(\sigma)$ denote the length of the longest increasing and decreasing run of σ, respectively. Furthermore, let $\text{dec}_d(\sigma)$ and $\text{inc}_d(\sigma)$ denote the length of the longest runs under d-noise. In Figure 1, we have $\text{inc}(\sigma) = 4$ because of $(7, 8, 10, 12)$ or $(7, 8, 13, 14)$ and $\text{dec}(\sigma) = 4$ because of $(7, 3, 2, 1)$.

Since every root-to-leaf path can be divided into an increasing and a decreasing run, we immediately obtain the following lemma.

Lemma 4. *For every sequence σ and all d we have*

$$\text{height}(\sigma) \leq \text{dec}(\sigma) + \text{inc}(\sigma),$$
$$\mathbb{E}\big(\text{height}_d(\sigma)\big) \leq \mathbb{E}\big(\text{dec}_d(\sigma) + \text{inc}_d(\sigma)\big).$$

In terms of upper bounds, $\text{dec}(\sigma)$ and $\text{inc}(\sigma)$ as well as $\text{dec}_d(\sigma)$ and $\text{inc}_d(\sigma)$ behave equally. The reason is that given a sequence σ, the sequence τ with $\tau_i = 1 - \sigma_i$ has the properties $\text{dec}(\sigma) = \text{inc}(\tau)$ and $\mathbb{E}\big(\text{dec}_d(\sigma)\big) = \mathbb{E}\big(\text{inc}_d(\tau)\big)$. This observation together with Lemma 4 proves the next lemma.

Lemma 5. *For all d, we have*

$$\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{height}_d(\sigma)\big) \leq 2 \cdot \max_{\sigma \in [0,1]^n} \mathbb{E}\big(\text{inc}_d(\sigma)\big).$$

The lemma states that in order to bound the smoothed height of search trees from above we can instead bound the smoothed length of increasing or decreasing runs. To simplify the analysis even further, we show that we can once more restrict our attention to sorted sequences.

Lemma 6. *For every σ and its sorted version τ, $\mathbb{E}\big(\mathrm{inc}_d(\sigma)\big) \leq \mathbb{E}\big(\mathrm{inc}_d(\tau)\big)$.*

4.2 Upper Bound on the Smoothed Height of Binary Search Trees

We are now ready to prove the upper bound for binary search trees by proving an upper bound on the smoothed length of increasing runs of sorted sequences. For this, we prove four lemmas, the last of which claims exactly the desired upper bound. Lemma 7 deals with $d = 1$ and states that $\mathbb{E}\big(\mathrm{height}_1(\sigma)\big) \in O(\sqrt{n})$ for every sequence σ. Lemma 8 states that in order to bound tree heights, we can divide sequences into (possibly overlapping) parts and consider the height of the trees induced by the subsequences individually. A less general form of the lemma has already been shown by Manthey and Reischuk [10, Lemma 4.1]. Lemma 9 establishes that if $d = n/\log^2 n$, a perturbed sequence behaves the same way as a completely random sequence with respect to the smoothed length of its longest increasing run. The core idea is to partition the sequence into a set of "completely random" elements, which behave as expected, and two sets of more bothersome elements lying in a small range. As we will see, the number of bothersome elements is roughly $\log^2 n$ and since the range of values of these elements is small, we can use the result $\mathbb{E}\big(\mathrm{height}_1(\sigma)\big) \in O(\sqrt{n})$ to show that their contribution to the length on an increasing run is just $O(\log n)$. Finally, in Lemma 10 we allow general $d \geq 1/n$. This case turns out to be reducible to the case $d = n/\log^2 n$ by scaling the numbers according to Lemma 1.

For the proofs of the lemmas, two technical terms will be helpful: For a given real interval $I = [a, b]$, we say that a position i of σ is *eligible* for I if $\overline{\sigma}_i$ can assume any value in I. In other words, i is eligible for $[a, b]$ if $\sigma_i \leq a$ and $\sigma_i + d \geq b$. Furthermore, we say that i is *regular* if $\overline{\sigma}_i$ actually lies inside I.

Lemma 7. *For all σ, we have $\mathbb{E}\big(\mathrm{inc}_1(\sigma)\big) \in O(\sqrt{n})$.*

Lemma 8. *For every sequence σ, all d, and every covering $U_1, U_2, \ldots, U_k$ of $\{1, \ldots, n\}$ (which means $\bigcup_{i=1}^k U_i = \{1, \ldots, n\}$), we have*

$$\mathrm{height}(\sigma) \leq \textstyle\sum_{i=1}^k \mathrm{height}(\sigma_{U_i}),$$
$$\mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \leq \textstyle\sum_{i=1}^k \mathbb{E}\big(\mathrm{height}_d(\sigma_{U_i})\big).$$

Lemma 9. *For all sequences σ, and $d = n/\log^2 n$, we have $\mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \in O(\log n)$.*

Lemma 10. *For every sequence σ and all $d \geq 1/n$ we have $\mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \in O\big(\sqrt{n/d} + \log n\big)$.*

Proof. If $d \in \Omega\big(n/\log^2 n\big)$, then $\mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \in O(\log n)$ by Lemma 9.

To prove the theorem for smaller values of d, we divide the sequence into subsequences. Let N be the largest real root of the equation $N^2/\log^2 N = nd$. Then $\log N \in \Theta(\log(nd))$, and thus $N = c \cdot \sqrt{nd} \cdot \log(nd)$ for some $c \in \Theta(1)$. Let n_j be the number of elements of σ with $\sigma_i \in [(j-1) \cdot N/n, j \cdot N/n]$. Choose $k_j \in \mathbb{N}$ such that $(k_j - 1) \cdot N < n_j \leq k_j N$. We divide the n_j elements of the interval $[(j-1) \cdot N/n, j \cdot N/n]$ into k_j subsequences $\sigma^{j,1}, \ldots, \sigma^{j,k_j}$ such that no subsequence contains more than N elements. Since

$$\textstyle\sum_{j=1}^{n/N} k_j \leq \sum_{j=1}^{n/N} \frac{n_j + N}{N} \leq 2n/N,$$

we obtain at most $2n/N$ such subsequences. Each subsequence spans at most an interval of length N/n and contains at most N elements. Thus, by Lemma 9, we have $\mathbb{E}\big(\mathrm{height}_d(\sigma^{j,\ell})\big) \in O(\log(N))$. Finally, Lemma 8 yields

$$\mathbb{E}\big(\mathrm{height}_d(\sigma)\big) \leq \textstyle\sum_{j=1}^{n/N} \sum_{\ell=1}^{k_j} \mathbb{E}\big(\mathrm{height}_d(\sigma^{j,\ell})\big) \in O\left(\frac{n \log N}{N}\right) = O\left(\sqrt{n/d}\right).$$

□

5 Smoothed Number of Quicksort Comparisons

In this section, we apply our results on binary search trees and left-to-right maxima to the performance of the quicksort algorithm. The following theorem summarizes the findings.

Theorem 3. *For $d \geq 1/n$ we have*

$$\max_{\sigma \in [0,1]^n} \mathbb{E}\big(\mathrm{qs}_d(\sigma)\big) \in \Theta\big(\tfrac{n}{d+1}\sqrt{n/d} + n \log n\big).$$

In other words, for $d \in O(1)$, we have at most $O(n\sqrt{n/d})$ comparisons, while for $d \in \Omega(1)$, we have at most $O(\frac{n}{d}\sqrt{n/d})$ comparisons. This means that d has a stronger influence for $d \in \Omega(1)$.

To prove the upper bound, we first need a lemma similar to Lemma 8 that allows us to estimate the number of comparisons of subsequences.

Lemma 11. *For every sequence σ, all d, and every covering $U_1, U_2, \ldots, U_k$ of $\{1, \ldots, n\}$, we have*

$$\mathrm{qs}(\sigma) \leq \textstyle\sum_{i=1}^{k} \mathrm{qs}(\sigma_{U_i}) + Q,$$
$$\mathbb{E}\big(\mathrm{qs}_d(\sigma)\big) = \mathbb{E}\big(\mathrm{qs}(\overline{\sigma})\big) \leq \textstyle\sum_{i=1}^{k} \mathbb{E}\big(\mathrm{qs}(\overline{\sigma}_{U_i})\big) + \mathbb{E}(\overline{Q}),$$

where Q is the number of comparisons of elements of σ_{U_i} with elements of $\sigma_{\{1,\ldots,n\}\setminus U_i}$ for any i and the random variable $\overline{Q}$ is defined analogously for $\overline{\sigma}$.

Lemma 12. *For every sequence σ and all $d \geq 1/n$, we have $\mathbb{E}\big(\mathrm{qs}_d(\sigma)\big) \in O\big(\frac{n}{d+1}\sqrt{n/d} + n \log n\big)$.*

Our upper bound is tight. The standard sorted sequence provides a worst case, but we use a sequence that is slightly easier to handle technically.

Lemma 13. *For $\sigma = (1/n, 2/n, \ldots, \frac{n}{2}/n, 1, 1, \ldots, 1)$ and all $d \geq 1/n$, we have $\mathbb{E}\big(\mathrm{qs}_d(\sigma)\big) \in \Omega\big(\frac{n}{d+1}\sqrt{n/d} + n\log n\big)$.*

6 Sorting Decreases the Number of Left-to-right Maxima

Lemma 2 states that sorting a sequence can never decrease the expected number of left-to-right maxima – at least when the noise is drawn uniformly from a single interval. Intuitively, this should not only hold for this kind of noise, but for *any* kind of noise – at least if the noise distribution is reasonably well-behaved. We demonstrate that this intuition is wrong and there exist a simple distribution and a sequence for which the sorted version has a lower expected number of left-to-right maxima than the original sequence.

Theorem 4. *The exist a sequence σ and a symmetric probability distribution $f\colon \mathbb{R} \to \mathbb{R}$ that is monotonically decreasing on $\mathbb{R}_+$ such that sorting σ to obtain τ decreases the expected number of left-to-right maxima after perturbation.*

To prove the theorem, we use the sequence $\sigma = (0, \ldots, 0, 1 + \frac{1}{\varepsilon}, \frac{1}{\varepsilon})$. The probability distribution has most of its mass in the interval $[-1, 1]$ with tails of length $1/\varepsilon$ to either side: $f(x) = 1 - 2\varepsilon$ for $x \in [-1, 1]$ and $f(x) = \varepsilon^2$ for $|x| \in [1, 1 + 1/\varepsilon]$. This distribution can easily be made smooth without changing the claim of the theorem.

7 Conclusion

The smoothed height of binary search trees and also the smoothed number of left-to-right maxima are $\Theta(\sqrt{n/d} + \log n)$; the smoothed number of quicksort comparisons is $\Theta(\frac{n}{d+1}\sqrt{n/d} + n\log n)$. While we obtain the average-case height of $\Theta(\log n)$ for binary search trees only for $d \in \Omega(n/\log^2 n)$ – which is large compared to the interval size $[0, 1]$ from which the numbers are drawn –, for the quicksort algorithm $d \in \Omega\big(\sqrt[3]{n/\log^2 n}\big)$ suffices so that the expected number of comparisons equals the average-case number of $\Theta(n \log n)$. On the other hand, the recursion depth of quicksort, which is equal to the tree height, can be as large as $\Omega\big(\sqrt{n/d}\big)$. Thus, although the average number of comparisons is already reached at $d \in \Omega\big(\sqrt[3]{n/\log^2 n}\big)$, the recursion depth remains asymptotically larger than its average value for $d \in o\big(n/(\log n)^2\big)$.

A natural question arising from our results is, what happens when the noise is drawn according to distributions other than the uniform distribution? As we have demonstrated, we cannot expect all distributions to behave in the same way as the uniform distribution. A natural candidate for closer examination is the normal distribution, for which first results on left-to-right maxima have already been obtained [4]. We conjecture that if $\max_{x\in\mathbb{R}} f(x) = \phi$, where f is the noise distribution, then the expected tree height and the expected number of left-to-right maxima are $O(\sqrt{n\phi} + \log n)$ while the expected number of quicksort comparisons is $O\big(\frac{\phi n}{\phi+1}\sqrt{n\phi}+n\log n\big)$. These bounds would be in compliance with our bounds for uniformly distributed noise, where $\phi = 1/d$.

In our study of the quicksort algorithm we used the first element as the pivot element. This choice simplifies the analysis but one would often use the median of the first, middle, and last element. Nevertheless, we conjecture that the same bounds as for the simple pivot strategy also hold for this pivot strategy.

References

1. Cyril Banderier, René Beier, Kurt Mehlhorn. Smoothed analysis of three combinatorial problems. In *Proc. 28th Int. Symp. on Math. Found. of Comput. Sci. (MFCS)*, vol. 2747 of *Lecture Notes in Comput. Sci.*, pp. 198–207. Springer, 2003.
2. Julien Basch, Leonidas J. Guibas, John Hershberger. Data structures for mobile data. *J. Algorithms*, 31(1):1–28, 1999.
3. Valentina Damerow. *Average and Smoothed Complexity of Geometric Structures.* PhD thesis, Universität Paderborn, 2006.
4. Valentina Damerow, Friedhelm Meyer auf der Heide, Harald Räcke, Christian Scheideler, Christian Sohler. Smoothed motion complexity. In *Proc. 11th Ann. European Symp. on Algorithms (ESA)*, vol. 2832 of *Lecture Notes in Comput. Sci.*, pp. 161–171. Springer, 2003.
5. Valentina Damerow, Christian Sohler. Extreme points under random noise. In *Proc. 12th Ann. European Symp. on Algorithms (ESA)*, vol. 3221 of *Lecture Notes in Comput. Sci.*, pp. 264–274. Springer, 2004.
6. Michael Drmota. An analytic approach to the height of binary search trees II. *J. ACM*, 50(3):333–374, 2003.
7. Michael Drmota. Profile and height of random binary search trees. *J. Iranian Statistical Society*, 3(2):117–138, 2004.
8. James Allen Fill, Svante Janson. Quicksort asymptotics. *J. Algorithms*, 44(1):4–28, 2002.
9. Donald E. Knuth. *Sorting and Searching*, vol. 3 of *The Art of Computer Programming.* Addison-Wesley, 2nd edition, 1998.
10. Bodo Manthey, Rüdiger Reischuk. Smoothed analysis of binary search trees. *Theoret. Comput. Sci.*, 378(3):292–315, 2007.
11. Bodo Manthey and Till Tantau. Smoothed analysis of binary search trees and quicksort under additive noise. Report 07-039, Electronic Colloquium on Computational Complexity (ECCC), 2007.
12. Bruce Reed. The height of a random binary search tree. *J. ACM*, 50(3):306–332, 2003.
13. Robert Sedgewick. The analysis of quicksort programs. *Acta Inform.*, 7(4):327–355, 1977.
14. Daniel A. Spielman, Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.
15. Daniel A. Spielman, Shang-Hua Teng. Smoothed analysis of algorithms and heuristics: Progress and open questions. In *Foundations of Computational Mathematics, Santander 2005*, pp. 274–342. Cambridge University Press, 2006.