On Smoothed Analysis of Quicksort and Hoare's Find*

Mahmoud Fouz[1] Manfred Kufleitner[2]
Bodo Manthey[3] Nima Zeini Jahromi[1]

[1] Saarland University, Department of Computer Science
Postfach 151150, 66041 Saarbrücken, Germany
mfouz@cs.uni-saarland.de, nzeini@studcs.uni-saarland.de

[2] Universität Stuttgart, FMI
Universitätsstraße 38, 70569 Stuttgart, Germany
manfred.kufleitner@fmi.uni-stuttgart.de

[3] University of Twente, Department of Applied Mathematics
Postbus 217, 7500 AE Enschede, The Netherlands
b.manthey@utwente.nl

We provide a smoothed analysis of Hoare's find algorithm, and we revisit the smoothed analysis of quicksort. Hoare's find algorithm – often called quickselect or one-sided quicksort – is an easy-to-implement algorithm for finding the k-th smallest element of a sequence. While the worst-case number of comparisons that Hoare's find needs is $\Theta(n^2)$, the average-case number is $\Theta(n)$. We analyze what happens between these two extremes by providing a smoothed analysis.

In the first perturbation model, an adversary specifies a sequence of n numbers of $[0,1]$, and then, to each number of the sequence, we add a random number drawn independently from the interval $[0,d]$. We prove that Hoare's find needs $\Theta\big(\frac{n}{d+1}\sqrt{n/d}+n\big)$ comparisons in expectation if the adversary may also specify the target element (even after seeing the perturbed sequence) and slightly fewer comparisons for finding the median.

In the second perturbation model, each element is marked with a probability of p, and then a random permutation is applied to the marked elements. We prove that the expected number of comparisons to find the median is $\Omega\big((1-p)\frac{n}{p}\log n\big)$.

Finally, we provide lower bounds for the smoothed number of comparisons of quicksort and Hoare's find for the median-of-three pivot rule, which usually yields faster algorithms than always selecting the first element: The pivot is the median of the first, middle, and last element of the sequence. We show that median-of-three does not yield a significant improvement over the classic rule.

[*] An extended abstract of this paper has appeared in the Proceedings of the 15th International Computing and Combinatorics Conference (COCOON 2009) [14].

to appear in *Algorithmica*

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* [30]. Smoothed analysis interpolates between average-case and worst-case analysis: Instead of taking a worst-case instance, we analyze the expected worst-case running time subject to slight random perturbations. The stronger the perturbation, the closer we come to the average case; if the perturbation is very weak, we get worst-case analysis.

In practice, neither can we assume that all instances are equally likely, nor that instances are precisely worst-case instances. The goal of smoothed analysis is to capture the notion of a *typical* instance mathematically. Typical instances are, in contrast to worst-case instances, often subject to measurement or rounding errors. Even if one assumes that nature is adversarial and that the instance at hand is initially a worst-case instance, due to such errors we would probably get a less difficult instance. On the other hand, typical instances still have some (adversarial) structure, which instances drawn completely at random do not. Since its invention, smoothed analysis has been applied successfully to a variety of different algorithms and problems [2,3,5–8,12,21,22,25]. Spielman and Teng [31] give a survey of results and open problems in smoothed analysis.

In this paper, we provide a smoothed analysis of Hoare's find [16] (see also Aho et al. [1, Algorithm 3.7]). Hoare's find, also called quickselect or one-sided quicksort, is a simple algorithm for finding the k-th smallest element of a sequence of numbers: Pick the first element as the pivot and compare it to all $n-1$ remaining elements. Assume that $\ell-1$ elements are smaller than the pivot. If $\ell = k$, then the pivot is the element that we are looking for. If $\ell > k$, then we call the algorithm recursively to find the k-th smallest element of the list of the smaller elements. If $\ell < k$, then we call the algorithm recursively to find the $(k-\ell)$-th smallest element among the larger elements. The number of comparisons to find the specified element is $\Theta(n^2)$ in the worst case and $\Theta(n)$ on average. These bounds hold for all $k \in \{1, 2, \ldots, n\}$. Furthermore, the variance of the number of comparisons is $\Theta(n^2)$ [17]. Neininger [24] gives a more thorough discussion of quickselect. As our first result, we close the gap between the quadratic worst-case running-time and the expected linear running-time by providing a smoothed analysis.

Hoare's find is closely related to quicksort [15] (see also Aho et al. [1, Section 3.5]), which needs $\Theta(n^2)$ comparisons in the worst case and $\Theta(n \log n)$ on average [19, Section 5.2.2]. The smoothed number of comparisons that quicksort needs has already been analyzed under different models [4,10]. Choosing the first element as the pivot element, however, results in poor running-time if the sequence is nearly sorted. There are two common approaches to circumvent this problem: First, one can choose the pivot randomly among the elements. However, randomness is needed to do so, which is sometimes expensive. Second, without any randomness, a common approach to circumvent this problem is to compute the median of the first, middle, and last element of the sequence and then to use this median as the pivot [28,29]. This method is faster in practice since it yields more balanced partitions and makes the worst-case behavior much more unlikely [19, Section 5.5]. It is also faster in both average and worst case, but only by constant factors [13,27]. Quicksort with the median-of-three rule is widely used, for instance in the `qsort()` implementation in the GNU standard C library `glibc` [26] and also in a recent very efficient implementation of quicksort on a GPU [9]. The median-of-three rule has also been used for Hoare's find, and the expected number of comparisons has been analyzed precisely [18]. Our second goal is a smoothed analysis of both quicksort and Hoare's find with the median-of-three rule to get a thorough understanding of this variant of these two algorithms.

1.1 Preliminaries

We denote *sequences* of real numbers by $s = (s_1, \ldots, s_n)$, where $s_i \in \mathbb{R}$. For $n \in \mathbb{N}$, we set $[n] = \{1, \ldots, n\}$. Let $U = \{i_1, \ldots, i_\ell\} \subseteq [n]$ with $i_1 < i_2 < \ldots < i_\ell$. Then $s_U = (s_{i_1}, s_{i_2}, \ldots, s_{i_\ell})$ denotes the *subsequence* of s of the elements at positions in U. We denote probabilities by $\mathbb{P}$ and expected values by $\mathbb{E}$.

Throughout the paper, log is the logarithm to base 2 and ln is the logarithm to base e. Furthermore, $\exp(x)$ denotes e^x.

We will assume for the sake of clarity that numbers of elements with specific properties, which are sometimes functions of parameters like n and d, are integers, and omit the tedious floor and ceiling functions that are actually necessary. Since we are interested in asymptotic bounds, the proofs remain valid.

Pivot Rules. Given a sequence s, a *pivot rule* determines one element of s as the *pivot element*. The pivot element will be the one to which we compare all other elements of s. In this paper, we consider four pivot rules, the last two of which play only an auxiliary role (the acronyms of the rules are in parentheses):

Classic rule (c): The first element s_1 of s is the pivot element.

Median-of-three rule (med3): The median of the first, middle, and last element is the pivot element, i.e., $\text{median}(s_1, s_{\lceil n/2 \rceil}, s_n)$.

Maximum-of-two rule (max2): The maximum of the first and the last element becomes the pivot element, i.e., $\max(s_1, s_n)$.

Minimum-of-two rule (min2): The minimum of the first and the last element becomes the pivot element, i.e., $\min(s_1, s_n)$.

The first pivot rule is the easiest-to-analyze and easiest-to-implement pivot rule. Its major drawback is that it yields poor running-times of quicksort and Hoare's find for nearly sorted sequences. The advantages of the median-of-three rule has already been discussed above. The last two rules are only used as tools for analyzing the median-of-three rule.

Quicksort, Hoare's Find, Left-to-right Maxima. Let s be a sequence of length n consisting of pairwise distinct numbers. Let p be the pivot element of s according to some rule. For the following definitions, let $L = \{i \in [n] \mid s_i < p\}$ be the set of positions of elements smaller than the pivot, and let $R = \{i \in [n] \mid s_i > p\}$ be the set of positions of elements greater than the pivot.

Quicksort is the following sorting algorithm: Given s, we construct s_L and s_R by comparing all elements to the pivot p. It is important for our analyses that the elements in s_L and in s_R are in the same order as in s. (In practical implementations of quicksort, this is not always fulfilled.) Then we sort s_L and s_R recursively to obtain s'_L and s'_R, respectively. Finally, we output $s' = (s'_L, p, s'_R)$. The number $\text{sort}(s)$ of comparisons needed to sort s is thus $\text{sort}(s) = (n-1) + \text{sort}(s_L) + \text{sort}(s_R)$ if s has a length of $n \geq 1$, and $\text{sort}(s) = 0$ if s is the empty sequence. We do not count the number of comparisons needed to find the pivot element. Since this number is $O(1)$ per recursive call for the pivot rules considered here, the asymptotics are not changed.

Hoare's find aims at finding the k-th smallest element of s. Let $\ell = |s_L|+1$. If $\ell = k$, then p is the k-th smallest element. If $\ell > k$, then we search for the k-th smallest element of s_L. If $\ell < k$, then we search for the $(k-\ell)$-th smallest element of s_R. Let $\text{find}(s,k)$ denote the number of comparisons needed to find the k-th smallest element of s, and let $\text{find}(s) = \max_{k \in [n]} \text{find}(s,k)$. As for quicksort, it is important for our analyses that the order of elements in s_L and s_R is the same as in s.

The number of *scan maxima* of s is the number of maxima seen when scanning s according to some pivot rule. This means that it is the number of pivot elements that Hoare's find requires to find the maximum element. Formally, let $\text{scan}(s) = 1 + \text{scan}(s_R)$, and let $\text{scan}(s) = 0$ if s is the empty sequence. If we use the classic pivot rule, the number of scan maxima is just the number of *left-to-right maxima*, i.e., the number of new maxima that we see if we scan s from left to right. Thus, scan maxima generalize left-to-right maxima to general pivot rules. The number of scan maxima is a useful tool for analyzing quicksort and Hoare's find, and has applications, e.g., in motion complexity [10].

We write $\text{c-scan}(s)$, $\text{med3-scan}(s)$, $\text{max2-scan}(s)$, and $\text{min2-scan}(s)$ to denote the number of scan maxima according to the classic, median-of-three, maximum, and minimum pivot rule, respectively. Similar notation is used for quicksort and Hoare's find.

Perturbation Model: Additive Noise. The first perturbation model that we consider is *additive noise*. Let $d > 0$. Given a sequence $s \in [0,1]^n$, i.e., the numbers $s_1, \ldots, s_n$ lie in the interval $[0,1]$, we obtain the perturbed sequence $\overline{s} = (\overline{s}_1, \ldots, \overline{s}_n)$ by drawing $\nu_1, \ldots, \nu_n$ uniformly and independently from the interval $[0,d]$ and setting $\overline{s}_i = s_i + \nu_i$. Note that $d = d(n)$ may be a function of the number n of elements, although this will not always be mentioned explicitly in the following.

We denote by $\text{scan}_d(s)$, $\text{sort}_d(s)$ and $\text{find}_d(s)$ the random number of scan maxima, quicksort comparisons, and comparisons of Hoare's find of $\overline{s}$. If needed, they are preceded by the acronym of the pivot rule used.

Our goal is to prove bounds for the smoothed number of comparisons that Hoare's find needs, i.e., $\max_{s \in [0,1]^n} \mathbb{E}\big(\text{c-find}_d(s)\big)$, as well as for Hoare's find and quicksort with the median-of-three pivot rule, i.e., $\max_{s \in [0,1]^n} \mathbb{E}\big(\text{med3-find}_d(s)\big)$ and $\max_{s \in [0,1]^n} \mathbb{E}\big(\text{med3-sort}_d(s)\big)$. Taking the maximum over all sequences reflects that the sequence s is chosen by an adversary.

If $d < 1/n$, the sequence s can be chosen such that the order of the elements is unaffected by the perturbation. In that case, smoothed analysis amounts to a worst case analysis. Thus, in the following, we assume $d \geq 1/n$. If d is large, the noise will swamp out the original instance, and the order of the elements of $\overline{s}$ will basically depend only on the noise. In that case, smoothed analysis amounts to an average case analysis. For intermediate d, we interpolate between these two extremes.

The choice of the intervals for the adversarial part and the noise is arbitrary. All that matters is the ratio of the sizes of the intervals: For $a < b$, we have $\max_{s \in [a,b]^n} \mathbb{E}\big(\text{find}_{d \cdot (b-a)}(s)\big) = \max_{s \in [0,1]^n} \mathbb{E}\big(\text{find}_d(s)\big)$. In other words, we can scale and shift the intervals, and the results depend only on the ratio of $b-a$ and d as well as the number of elements. The same holds for all other measures that we consider. We will exploit this in the analysis of Hoare's find.

Perturbation Model: Partial Permutations. The second perturbation model that we consider is *partial permutations*, introduced by Banderier et al. [4]. Here, the elements themselves are left unchanged. Instead, we randomly permute a random subsets of the elements.

Without loss of generality, we can assume that s is a permutation of a set of n numbers, say, $[n]$. The perturbation parameter is $p \in [0,1]$. Any element s_i (or, equivalently, any position i) is marked independently of the others with a probability of p. After that, all marked positions are randomly permuted: Let M be the set of positions that are marked, and let $\pi : M \to M$ be a permutation drawn uniformly at random. Then

$$\overline{s}_i = \begin{cases} s_{\pi(i)} & \text{if } i \in M \text{ and} \\ s_i & \text{otherwise.} \end{cases}$$

If $p = 0$, no element is marked, and we obtain worst-case bounds. If $p = 1$, all elements are marked, and $\overline{s}$ is a uniformly drawn random permutation. We denote by $\text{pp-find}_p(s)$ the random number of comparisons that Hoare's find needs with the classic pivot rule when s is perturbed.

1.2 Known Results

Additive noise is perhaps the most basic and natural perturbation model for smoothed analysis. In particular, Spielman and Teng added random numbers to the entries of the adversarial matrix in their smoothed analysis of the simplex algorithm [30]. Damerow et al. [10] analyzed the smoothed number of left-to-right maxima of a sequence under additive noise. They proved a tight bound of

$$\max_{s \in [0,1]^n} \mathbb{E}\big(\text{c-scan}_d(s)\big) \in \Theta\big(\sqrt{n/d} + \log n\big). \tag{1}$$

Moreover, they proved that the same bound also holds for the smoothed height of binary search trees. Finally, they also proved a tight bound for quicksort, namely

$$\max_{s \in [0,1]^n} \mathbb{E}\big(\text{c-sort}_d(s)\big) \in \Theta\left(\frac{n}{d+1} \cdot \sqrt{n/d}\right).$$

Banderier et al. [4] introduced partial permutations as a perturbation model for ordering problems. They proved that a sequence of n numbers has, after partial permutation, an expected number of $O(\sqrt{(n/p)\log n})$ left-to-right maxima, and they proved a lower bound of $\Omega(\sqrt{n/p})$ for $p \leq \frac{1}{2}$. This has later been tightened to

$$\max_s \mathbb{E}\big(\text{pp-ltrm}_p(s)\big) \in \Theta\big((1-p) \cdot \sqrt{n/p}\big)$$

and generalized to binary search trees, for which the same bounds hold [20]. Banderier et al. [4] also analyzed quicksort, for which they proved an upper bound of

$$\max_s \mathbb{E}\big(\text{pp-sort}_p(s)\big) \in O\left(\frac{n}{p} \log n\right).$$

1.3 New Results

First, we give a smoothed analysis of Hoare's find under additive noise. We consider both finding an arbitrary element and finding the median. In the first case, the adversary specifies

	$d \leq 1/2$	$d \in (1/2, 2)$	$d = 2$	$d > 2$	
quicksort					
c	$\Theta\big(n\sqrt{n/d}\big)$	$\Theta\big(n^{3/2}\big)$	$\Theta\big(n^{3/2}\big)$	$\Theta\big((n/d)^{3/2}\big)$	[10]
med3	$\Omega\big(n\sqrt{n/d}\big)$	$\Omega\big(n^{3/2}\big)$	$\Omega\big(n^{3/2}\big)$	$\Omega\big((n/d)^{3/2}\big)$	Cor. 5.2
Hoare's find					
median, c	$\Theta\big(n\sqrt{n/d}\big)$	$\Omega\big(n^{3/2}(1 - \sqrt{d/2})\big)$	$\Theta(n \log n)$	$O\big(\frac{d}{d-2} \cdot n\big)$	Thm. 3.1
general, c	$\Theta\big(n\sqrt{n/d}\big)$	$\Theta\big(n^{3/2}\big)$	$\Theta\big(n^{3/2}\big)$	$\Theta\big((n/d)^{3/2}\big)$	Thm. 2.1
general, med3	$\Omega\big(n\sqrt{n/d}\big)$	$\Omega\big(n^{3/2}\big)$	$\Omega\big(n^{3/2}\big)$	$\Omega\big((n/d)^{3/2}\big)$	Thm. 5.1
scan maxima					
c	$\Theta\big(\sqrt{n/d}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n/d}\big)$	[10]
med3	$\Theta\big(\sqrt{n/d}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n/d}\big)$	Thm. 6.1
binary search trees					
	$\Theta\big(\sqrt{n/d}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n}\big)$	$\Theta\big(\sqrt{n/d}\big)$	[10]

Table 1: Bounds for additive noise. The upper bound for Hoare's find (general, classic) for $d \in (1/2, 2)$ applies also to Hoare's find for finding the median. Note that, even for large d, the bounds for quicksort, Hoare's find, and scan maxima never drop below $\Omega(n \log n)$, $\Omega(n)$, and $\Omega(\log n)$, respectively.

k, and we have to find the k-th smallest element (Section 2). We prove tight bounds of $\Theta\big(\frac{n}{d+1}\sqrt{n/d} + n\big)$ for the expected number of comparisons. This means that already for very small $d \in \omega(1/n)$, the smoothed number of comparisons is reduced asymptotically compared to the worst case of $O(n^2)$ comparisons. If d is a small constant, i.e., the noise is a small percentage of the data values like 1%, then $O(n^{3/2})$ comparisons suffice.

If the adversary is to choose k, our lower bound suggests that we will have either $k = 1$ or $k = n$. However, the main task of Hoare's find is to find medians. Thus, second, we give a separate analysis of how many comparisons are needed to find the median (Section 3). Surprisingly, it turns out that under additive noise, finding medians is easier than finding maximums or minimums, in particular for large d: For $d \leq 1/2$, we have roughly the same bounds as above. For $d \in (\frac{1}{2}, 2)$, we prove a lower bound of $\Omega\big(n^{3/2} \cdot (1 - \sqrt{d/2})\big)$, which again matches the upper bound of Section 2 that of course still applies (Section 3.1). For $d > 2$, we prove that a linear number of comparisons suffices for finding the median, which is considerably less than the $\Omega\big((n/d)^{3/2}\big)$ general lower bound of Section 2 for this case. Thus, we have a phase transition at $d = 2$. For the special value $d = 2$, we prove a tight bound of $\Theta(n \log n)$ (Sections 3.3 and 3.4).

After that, we aim at analyzing the median-of-three rule. As a tool, we analyze the number of scan maxima under the maximum-of-two, minimum-of-two, and median-of-three rule (Section 4). We show that the same asymptotic bounds as for the classic rule carry over to these rules. Then we apply these findings to quicksort and Hoare's find (Section 5). Again, we prove a lower bound that matches the lower bound for the classic rule. Thus, the median-of-three rule does not improve the asymptotics under additive noise.

The results concerning additive noise are summarized in Table 1.

Finally, and to contrast our findings for additive noise, we analyze Hoare's find under partial permutations (Section 6). We prove that there exist sequences on which Hoare's find needs an

quicksort	$O\big((n/p)\log n\big)$	[4]
Hoare's find	$\Omega\big((1-p)(n/p)\log n\big)$	Thm. 6.1
scan maxima & binary search trees	$\Theta\big((1-p)\sqrt{n/p}\big)$	[4, 20]

Table 2: Overview of bounds for partial permutations. All results are for the classic pivot rule. The upper bound for quicksort also holds for Hoare's find, while the lower bound for Hoare's find also applies to quicksort.

expected number of $\Omega\big((1-p)\cdot\frac{n}{p}\cdot\log n\big)$ comparisons. Since this matches the upper bound for quicksort [4] up to a factor of $O(1-p)$, this lower bound is almost tight.

For completeness, Table 2 gives an overview of the results for partial permutations.

2 Smoothed Analysis of Hoare's Find: General Bounds

In this section, we prove tight bounds for the smoothed number of comparisons that Hoare's find needs using the classic pivot rule. We allow the adversary to specify the target element after the perturbation of the original sequence. The number of comparisons is maximized, at least asymptotically, when the target element is the maximum element. Thus, we analyze Hoare's find for the maximum element.

Theorem 2.1. *For $d \geq 1/n$, we have*

$$\max_{s\in[0,1]^n} \mathbb{E}\big(\text{c-find}_d(s)\big) \in \Theta\big(\tfrac{n}{d+1}\sqrt{n/d}+n\big).$$

The following subsection contains the proof of the upper bound. After that, we prove the lower bound.

2.1 General Upper Bound for Hoare's Find

We already have an upper bound for the smoothed number of comparisons that quicksort needs [10]. This bound is $O\big(\frac{n}{d+1}\cdot\sqrt{n/d}+n\log n\big)$, which matches the bound of Theorem 2.1 for $d \in O\big(n^{1/3}\cdot\log^{-2/3} n\big)$. We have $\text{find}(s) \leq \text{sort}(s)$ for any s. By monotonicity of the expectation, this inequality yields $\mathbb{E}\big(\text{find}_d(s)\big) \leq \mathbb{E}\big(\text{sort}_d(s)\big)$. So in the following we assume $d \in \Omega\big(n^{1/3}\cdot\log^{-2/3} n\big)$.

In the next lemma, we show how to analyze the number of comparisons in terms of subsequences. Lemma 2.3 states that adding a new target element to a sequence increases the number of comparisons at most by an additive $O(n)$. Lemma 2.4 states the actual upper bound.

Lemma 2.2. *Let s be a sequence, and let $k \in [n]$. Let j be the position of the k-th smallest element of s. Let $U_1,\ldots,U_m$ be a covering of $[n]$ (i.e., $\bigcup_{\ell=1}^m U_\ell = [n]$) such that $j \in U_\ell$ for all $\ell \in [m]$. Let $k_1,\ldots,k_m$ be chosen such that s_j is the k_ℓ-th smallest element of s_{U_ℓ}. Then*

$$\text{c-find}(s,k) \leq \sum_{\ell=1}^{m} \text{c-find}(s_{U_\ell},k_\ell) + Q,$$

where Q is the total number of comparisons of positions p and q during the execution of Hoare's find on s such that p and q do not share a common set in the covering, i.e., $\{p, q\} \not\subseteq U_\ell$ for all $\ell \in [m]$.

Proof. Fix any $\ell \in [m]$, and let a and b be two elements of s_{U_ℓ} that are not compared for finding the k_ℓ-th smallest element of U_ℓ. Without loss of generality, we assume that $a < b$ and that a appears before b in s_{U_ℓ} (and hence in s).

If a is not compared to b, then this is due to one of the following two reasons:

1. There is a c prior to a in s_{U_ℓ} such that either $s_{k_\ell} \leq c < a$ or $b < c \leq s_{k_\ell}$.

2. There is a c in s_{U_ℓ} prior to a with $a < c < b$.

In either case, a and b are also not compared while searching for the k-th smallest element of s. Hence, all comparisons are accounted for, either in a c-find(s_{U_ℓ}) or in Q, which proves the lemma. □

Lemma 2.3. *Let s be any sequence of length n, and let s' be obtained from s by inserting one arbitrary element t at an arbitrary position of s. Let t be the k-th smallest element of s'. Then*

$$\text{c-find}(s', k) \leq \text{c-find}(s) + n + O(1).$$

Proof. The number of comparisons to find t in s' is maximal if we insert t as the last element. Thus, it suffices to consider this case.

Consider the two binary search trees obtained from s' and s by inserting elements one after the other (without rotations or balancing). These two trees differ only by the former having t as a leaf. Let $\tilde{t}$ be the parent of t in the binary search tree of s'. The execution of Hoare's find to find t in s' or to find $\tilde{t}$ in s yields the same pivots, except for the last step, where we actually find t. The subsequences obtained during the execution are almost identical; they only differ by the element t. Since there are at most n pivots, this costs at most n comparisons more. Plus $O(1)$ comparisons for the last step yields the desired bound. □

With the two lemmas above, we are ready to prove the upper bound for $d \in \Omega\big(n^{1/3} \cdot \log^{-2/3} n\big)$.

Lemma 2.4. *Let $d \in \Omega(n^{1/3} \cdot \log^{-2/3} n)$, and let s be arbitrary. Then*

$$\mathbb{E}\big(\text{c-find}_d(s)\big) \in O\big((n/d)^{3/2} + n\big).$$

Proof. The key insight is the following observation: Given that an element $\overline{s}_i$ assumes a value in $[1, d]$, it is uniformly distributed in this interval.

Let $R = \{i \mid \overline{s}_i \in [1, d]\}$ be the set of all indices of *regular* elements, i.e., elements that are uniformly distributed in $[1, d]$. Let $F = \{i \mid \nu_i \leq 3\}$ be the set of all elements with noise at most 3, which covers in particular all i that are not in R due to $\overline{s}_i$ being too small. Analogously, let $B = \{i \mid \nu_i \geq d - 3\}$ be the set of all elements with noise at least $d - 3$, which includes all i that are not in R due to $\overline{s}_i$ being too large. We have $F \cup R \cup B = [n]$.

We prove that the expected values of c-find$_d(\overline{s}_F)$, c-find$_d(\overline{s}_R)$, c-find$_d(\overline{s}_B)$ as well as the expected number of comparisons between elements in different subsets are bounded from above by $O\big((n/d)^{3/2} + n\big)$. Combining Lemmas 2.2 and 2.3 yields the result. (Lemma 2.3 is necessary since we have to add the target element to all three sets.)

First, $\mathbb{E}\big(\text{c-find}_d(\overline{s}_R)\big) \in O(n) \subseteq O\big((n/d)^{3/2} + n\big)$ since the elements of $\overline{s}_R$ are uniformly distributed in $[1, d]$, and Hoare's find needs only a linear number of comparisons in this case [1, Theorem 3.11]. Second, $\mathbb{E}\big(\text{c-find}_d(\overline{s}_B)\big) = \mathbb{E}\big(\text{c-find}_d(\overline{s}_F)\big)$. This is because the distributions of both sequences is the same except that the values are shifted by $d-3$. Thus, we can restrict ourselves to analyzing $\mathbb{E}\big(\text{c-find}_d(\overline{s}_F)\big)$. Given that $i \in F$, the noise ν_i is uniformly distributed in $[0, 3]$. Thus, we can apply the upper bound for quicksort for $d = 3$, which is $O(|F|^{3/2})$ [10]. The probability that an element is in F is $\frac{3}{d}$. By Chernoff's bound [11], the probability that $|F| > \frac{6n}{d}$ is at most $\exp(-n^{\varepsilon})$ for some constant $\varepsilon > 0$. If this happens nevertheless, we bound the number of comparisons by the worst-case bound of $\Theta(n^2)$. Due to the small probability, this contributes only $o(1)$ to the expected value. If F contains at most $6n/d$ elements, then we obtain $\mathbb{E}\big(\text{c-find}(\overline{s})_F\big) \in O\big((n/d)^{3/2}\big)$.

Third, and finally, the number of comparisons between elements with $\overline{s}_i \leq 1$ and elements with $\nu_j \geq 3$ remains to be considered. Similarly, comparisons between elements with $\overline{s}_i \geq d-1$ and $\nu_j \leq d-3$ have to be considered. All other comparisons have already been counted. By symmetry, we can restrict ourselves to considering the former case only. In the first subcase, we count the number of comparisons with an element with $\overline{s}_i \leq 1$ being the pivot. We observe that $\overline{s}_i \leq 1$ is compared to $\overline{s}_j$ with $\nu_j \geq 3$ only if there is no position $\ell < i$ with $\nu_\ell \in [2, 3]$. For every element ℓ, we have $\mathbb{P}\big(\overline{s}_\ell \leq 1\big) = \frac{1-s_\ell}{d} \leq \frac{1}{d} = \mathbb{P}\big(\nu_\ell \in [2,3]\big)$. Thus, because of $\mathbb{P}\big(\overline{s}_\ell \leq 1\big) \leq \mathbb{P}\big(\nu_\ell \in [2,3]\big)$, the probability that we have m elements $i_1, \ldots, i_m$ with $\overline{s}_{i_z} \leq 1$ for $1 \leq z \leq m$ before the first position ℓ with $\nu_\ell \in [2, 3]$ is bounded from above by 2^{-m}. If we have that many elements, we bound the number of such comparisons by mn. Thus, an upper bound for the number of such comparisons is $\sum_{m \in \mathbb{N}} 2^{-m} mn \in O(n)$. Similarly, the number of comparisons between elements with $\overline{s}_i \leq 1$ and $\overline{s}_j \geq d$ (ignoring which of them is the pivot) is also $O(n)$.

In the second subcase, let us count the number of comparisons between one element with $\nu_j \geq 3$ and $\overline{s}_j \leq d$ and another element with $\overline{s}_i \leq 1$ with the former being the pivot. An upper bound for this is the number of comparisons of elements satisfying $\overline{s} \in [1, d]$ (which is just s'_R) with elements satisfying $\overline{s}_i \leq 1$. There are at most $O(n/d)$ of the latter with high probability by Chernoff's bound (otherwise, we bound the number of comparisons by n^2 again), and only left-to-right *minima* of $\overline{s}_R$ become pivot elements. The expected number of left-to-right minima of a random sequence is $O(\log n)$ [4,10], resulting in an $O\big(\frac{n \cdot \log n}{d}\big) \subseteq O(n)$ bound since $d \in \Omega(\log n)$. □

2.2 General Lower Bound for Hoare's Find

Now we turn to the general lower bound. The proof is similar to the lower bound proof for quicksort [10].

Lemma 2.5. *For the sequence $s = (1/n, 2/n, 3/n, \ldots, \frac{n}{2}/n, 1, 1, \ldots, 1)$ and all $d \geq 1/n$, we have*

$$\mathbb{E}\big(\text{c-find}_d(s)\big) \in \Omega\left(\frac{n}{d+1}\sqrt{n/d} + n\right).$$

Proof. We aim at finding the maximum element. Then the pivot elements are just the left-to-right maxima. As in the analysis of the smoothed number of quicksort comparisons, any left-to-right maximum $\overline{s}_i$ of $\overline{s}$ must be compared to every element of $\overline{s}$ that is greater than $\overline{s}_i$ with $\overline{s}_i$ being the pivot element. We have an expected number of $\Theta\big(\sqrt{n/d} + \log n\big) \subseteq \Omega(\sqrt{n/d})$ left-to-right maxima among the first $n/2$ elements of s [10].

If $d \le \frac{1}{2}$, then every element of the second half is greater than any element of the first half. In this case, an expected number of $\frac{n}{2} \cdot \Omega\bigl(\sqrt{n/d}\bigr) = \Omega\bigl(\frac{n}{d+1} \cdot \sqrt{n/d}\bigr)$ comparisons is needed.

If $d > \frac{1}{2}$, a sufficient condition that an element $\overline{s}_i$ $(i > n/2)$ is greater than all elements of the first half is $\nu_i > d - \frac{1}{2}$, which happens with a probability of $\frac{1}{2d}$. Thus, we expect to see $\frac{n}{4d}$ such elements. Since the number of left-to-right maxima in the first half and the number of elements $\overline{s}_i$ with $\nu_i > d - \frac{1}{2}$ in the second half are independent random variables, we can multiply their expected values to obtain a lower bound of $\Omega\bigl(\frac{n}{4d} \cdot \sqrt{n/d}\bigr)$. This is equal to $\Omega\bigl(\frac{n}{d+1} \cdot \sqrt{n/d}\bigr)$ as $d > \frac{1}{2}$.

Observing that $\mathbb{E}\bigl(\text{find}_d(s)\bigr)$ drops never below the best-case number of comparisons, which is $\Omega(n)$, completes the proof. □

3 Smoothed Analysis of Hoare's Find: Finding the Median

In this section, we prove tight bounds for the special case of finding the median of a sequence using Hoare's find. Surprisingly, finding the median seems to be easier: fewer comparisons suffice.

Theorem 3.1. *Depending on d, we have the following bounds for*

$$\max_{s \in [0,1]^n} \mathbb{E}\bigl(\text{c-find}_d(s, \lceil n/2 \rceil)\bigr) :$$

For $d \le \frac{1}{2}$, we have $\Theta\bigl(n \cdot \sqrt{n/d}\bigr)$. For constant $d \in (\frac{1}{2}, 2)$, we have $\Omega\bigl((1 - \sqrt{d/2}) \cdot n^{3/2}\bigr)$ and $O\bigl(n^{3/2}\bigr)$. For $d = 2$, we have $\Theta\bigl(n \cdot \log n\bigr)$. Finally, for $d > 2$, we have $O\bigl(\frac{d}{d-2} \cdot n\bigr)$.

The upper bounds of $O(n \cdot \sqrt{n/d})$ for $d \le \frac{1}{2}$ and $\frac{1}{2} < d < 2$ follow from our general upper bound (Theorem 2.1). For $d \le \frac{1}{2}$, our lower bound construction for the general bounds also works: The median is among the last $n/2$ elements, which are the big ones. (We might want to have $\lceil n/2 \rceil$ or $n/2 + 1$ large elements to assure this.) The rest of the proof remains the same.

For $d > 2$, Theorem 3.1 states a linear bound, which is asymptotically equal to the average-case bound of $O(n)$ [1, Theorem 3.11]. Thus, we do not need a lower bound in this case.

In the following sections, we give proofs for the remaining cases. First, we prove the lower bound for $\frac{1}{2} < d < 2$ (Section 3.1), then we prove the upper bound for $d > 2$ (Section 3.2). Finally, we prove the bound of $\Theta(n \log n)$ for $d = 2$ in Sections 3.3 and 3.4.

3.1 Lower Bound for $d < 2$

We will prove lower bounds matching our general upper bound of $O(\frac{n}{d+1} \cdot \sqrt{n/d})$. Since $d < 2$, this equals $O(n \cdot \sqrt{n/d})$. We already have a bound for $d \le \frac{1}{2}$, thus we can restrict ourselves to $\frac{1}{2} < d < 2$. The idea is similar to the lower bound construction for quicksort [10].

Lemma 3.2. *Let $\frac{1}{2} < d < 2$. Then there exists a family $(s^{(n)})_{n \in \mathbb{N}}$, where $s^{(n)}$ has a length of n, such that*

$$\mathbb{E}\bigl(\text{c-find}_d(s^{(n)}, \lceil n/2 \rceil)\bigr) \in \Omega\bigl((1 - \sqrt{d/2}) \cdot n^{3/2}\bigr).$$

Proof. Let

$$s = s^{(n)} = \bigl(\tfrac{1}{n}, \tfrac{2}{n}, \ldots, \tfrac{a}{n}, \underbrace{1, \ldots, 1}_{b \text{ elements}}\bigr)$$

with $a + b = n$, where a and b will be chosen later on. We will refer to the first a elements, which have values of $\frac{i}{n}$, as the small elements and to the last b elements, all of which are of value 1, as the large elements. A sufficient condition that a large element is greater than all small elements is that its noise is at least $d - 1 + \frac{a}{n}$. Thus, the probability that a particular large element is greater than all small elements in $\overline{s}$ is at least $(1 - \frac{a}{n})/d$. Hence, we expect to see at least $b(1 - \frac{a}{n})/d$ such elements. In order to get our lower bound, we want the median of $\overline{s}$ to be among the large elements. For that purpose, we need $b(1 - \frac{a}{n})/d \geq n/2$, which is equivalent to $b \geq \frac{nd}{2-2a/n} = \frac{n^2 d}{2n-2a} = \frac{n^2 d}{2b}$. Thus, we need $b \geq n \cdot \sqrt{d/2}$. (Since $b \leq n$, this requirement makes the construction impossible for $d \geq 2$.)

The number of large elements that are greater than all small elements is binomially distributed. Thus, with a probability that is bounded from below by a positive constant, at least $n/2$ of the large elements are greater than all small elements of $\overline{s}$. In this case, the median is among the large elements. Thus, every left-to-right maximum of the small elements has to be compared to at least $n/2$ elements. The lower bound for the number of left-to-right maxima under uniform noise [10] yields

$$\mathbb{E}\left(\text{c-scan}_d(\tfrac{1}{n}, \ldots, \tfrac{a}{n})\right) = \mathbb{E}\left(\text{c-scan}_{\frac{dn}{a}}(\tfrac{1}{a}, \ldots, \tfrac{a}{a})\right) \in \Omega\big(\sqrt{a^2/dn}\big),$$

which in turn gives us

$$\mathbb{E}\big(\text{c-find}_d(s, \lceil n/2 \rceil)\big) \in \Omega\left(\frac{\sqrt{a^2}}{\sqrt{dn}} \cdot \frac{n}{2}\right) = \Omega\big(a\sqrt{n}\big).$$

The constraint $b \geq n \cdot \sqrt{d/2}$ yields $a \leq n \cdot \big(1 - \sqrt{d/2}\big)$, which yields the result. □

3.2 Upper Bound for $d > 2$

In this section, we prove that the expected number of comparisons that Hoare's find needs in order to find the median is linear for any $d > 2$, with the constant factor depending on d.

First, we prove a crucial fact about the value of the median: Intuitively, the median should be around $d/2$ if all elements of s are 0, and it should be around $1 + d/2$ if all elements of s are 1. For arbitrary input sequences s, it should be between these two extremes. In other words: Independent of the input sequence, the median will be neither much smaller than $d/2$ nor much greater than $1 + d/2$ with high probability. This lemma will also be needed in Section 3.3, where we prove an upper bound for the case $d = 2$.

Lemma 3.3. *Let $s \in [0,1]^n$, and let $d > 0$. Let $\xi = c\sqrt{\log n/n}$. Let m be the median of $\overline{s}$. Then*

$$\mathbb{P}\left(m \notin \left[\frac{d}{2} - \xi, 1 + \frac{d}{2} + \xi\right]\right) \leq 2 \cdot n^{-2c^2/d^2}.$$

Proof. Let $b = \frac{d}{2} - \xi$. We restrict ourselves to prove $\mathbb{P}\big(m < b\big) \leq n^{-2c^2 \log n/d^2}$. The other bound follows by symmetry. Fix any i. The probability that $\overline{s}_i < b$ is $\max\{0, \frac{b - s_i}{d}\} \leq \frac{b}{d}$. If $m < b$, then at least $n/2$ elements must be smaller than b. The expected number of elements smaller than b is at most $\frac{bn}{d}$. We apply Chernoff's bound [11, Theorem 1.1] and obtain

$$\begin{aligned}\mathbb{P}\big(m < b\big) &= \mathbb{P}\big(\text{at least } n/2 \text{ elements are smaller than } b\big) \\ &< \exp\left(-\frac{2\left(\frac{n}{2} - \frac{bn}{d}\right)^2}{n}\right) = \exp\left(-\frac{2\xi^2 n}{d^2}\right) = \exp\left(-\frac{2c^2 \log n}{d^2}\right) = n^{-2c^2/d^2}.\end{aligned}$$

□

The idea to prove the upper bound for $d > 2$ is as follows: Since $d > 2$ and according to Lemma 3.3 above, it is likely that any element can assume a value greater or smaller than the median. Thus, after we have seen a few pivots (for which we "pay" with $O\bigl(\frac{d}{d-2} \cdot n\bigr)$ comparisons), all elements that are not already cut off are within some small interval around the median. These elements are uniformly distributed in this interval. Thus, the linear average-case bound applies.

Lemma 3.4. *Let $d > 2$ be bounded away from 2. Then*

$$\max_{s \in [0,1]^n} \mathbb{E}\bigl(\text{c-find}_d(s, \lceil n/2 \rceil)\bigr) \in O\left(\frac{d}{d-2} \cdot n\right).$$

Proof. We can assume that $d \in o(\sqrt{n/\log n})$: For larger values of d, we already have a linear bound by Theorem 2.1. Let $\xi = d\sqrt{\log n/n}$. By Lemma 3.3, the median of $\overline{s}$ falls into the interval $\bigl[\frac{d}{2} - \xi, 1 + \frac{d}{2} + \xi\bigr]$ with a probability of at least $1 - 2n^{-2}$. If the median does not fall into this interval, we bound the number of comparisons by the worst-case bound of $O(n^2)$, which contributes only $O(1)$ to the expected value.

The key observation to get the linear bound is the following: Every element of $\overline{s}$ can assume any value in the interval $[1, d]$. Thus, with a probability of at least $\frac{d/2-\xi-1}{d}$, it assumes a value smaller than the median but larger than 1 (called a *low cutter*). Similarly, with a probability of at least $\frac{d/2-\xi-1}{d}$, it assumes a value greater than the median but smaller than d (called a *high cutter*).

Now assume that we have already seen a low cutter a and a high cutter b. Then any element that remains to be considered is uniformly distributed in the interval $[a, b]$. Thus, the linear average-case bound for the number of comparisons applies [1, Theorem 3.11], and we expect to need only $O(n)$ additional comparisons.

Until we have seen both a low and a high cutter, we bound the number of comparisons per iteration by the trivial upper bound of n. Let c_ℓ be the position of the first low cutter and let c_h be the position of the first high cutter. Then, in this way, we get a bound of $\max(c_\ell, c_h) \cdot n + O(n)$. The expected values of c_ℓ and c_h remain to be bounded.

The probability that an element is a low cutter is at least $\frac{d/2-\xi-1}{d}$. Thus, the expected number of elements until we see a low cutter is at most $\frac{2d}{d-2\xi-2}$. The same applies to the high cutters. Hence,

$$\mathbb{E}\bigl(\max(c_\ell, c_h)\bigr) \le \mathbb{E}\bigl(c_\ell\bigr) + \mathbb{E}\bigl(c_h\bigr) \le \frac{4d}{d - 2\xi - 2} \in O\left(\frac{d}{d-2}\right).$$

The "$\in$" holds since $d \in o(\sqrt{n/\log n})$, which implies $\xi \in o(1)$. □

3.3 Upper Bound for $d = 2$

In this section, we prove that the expected number of comparisons for finding the median in case of $d = 2$ is $O(n \log n)$, which matches the lower bound of the next section. Before we dive into the actual proof, we will rule out two bad cases by showing that each of these bad cases occurs only with a probability of at most $O(1/n)$. If one of the bad events happens, then we bound the number of comparisons by the worst-case bound of $\Theta(n^2)$. This contributes only $O(n)$ to the expected value, which is negligible.

First, with a probability of at most $O(1/n)$, there is an interval of length $1/n$ that contains more than $\log n$ elements of the perturbed sequence. Second, with a probability of at most $O(n^{-2})$, the median is larger than 2, provided that there are more than $4\sqrt{n \log n}$ elements of the original (unperturbed) sequence s that are at most $1/2$.

Lemma 3.5. *Let $s \in [0,1]^n$. Then*

$$\mathbb{P}\left(\exists a \in [0, 3 - \tfrac{1}{n}] \text{ such that } |\{\bar{s}_i \in [a, a + \tfrac{1}{n}]\}| \geq \log n\right) \in O(1/n).$$

Proof. Let n be sufficiently large. We divide the interval $[0,3]$ into $3n$ intervals of length $1/n$. By a standard balls-into-bins argument (see, e.g., [23, Lemma 5.1]), the probability that there is an interval such that more than $O(\ln n / \ln \ln n)$ elements assume a value in this interval is $O(1/n)$. Since any interval $[a, a + \frac{1}{n}]$ intersects with at most two bins, the probability that there is an interval $[a, a + \frac{1}{n}]$ with more than $\log n$ elements is also $O(1/n)$. □

Lemma 3.6. *Let $d = 2$. Assume that the unperturbed sequence s contains at least $4\sqrt{n \log n}$ elements that are at most $1/2$. Then the probability that the median of the perturbed sequence is greater than 2 is at most $O(n^{-2})$.*

Proof. Let $\ell = 4\sqrt{n \log n}$ and assume that s contains at least ℓ elements that are at most $1/2$. Let X denote the number of elements in the perturbed sequence $\bar{s}$ that are larger than 2. Then

$$\mathbb{E}(X) \leq \tfrac{1}{2}(n - \ell) + \tfrac{1}{4}\ell = \tfrac{1}{2}n - \tfrac{1}{4}\ell.$$

Chernoff's bound [11, Theorem 1.1] yields

$$\begin{aligned} &\mathbb{P}(\text{median is larger than } 2) = \mathbb{P}(X \geq n/2) \\ &\leq \exp\left(\frac{-2(\ell/4)^2}{n}\right) = \exp\bigl(-2\log n\bigr) \in O(n^{-2}). \end{aligned}$$

□

We are now ready to prove the upper bound on the number of comparisons for $d = 2$.

Lemma 3.7. *We have*

$$\max_{s \in [0,1]^n} \mathbb{E}\bigl(\text{c-find}_2(s, \lceil n/2 \rceil)\bigr) \in O\left(n \log n\right).$$

Proof. By Lemmas 3.3, 3.5, and 3.6, the probability that any of the following events happens is at most $O(1/n)$:

1. The median of $\bar{s}$ does not belong to the interval $\bigl[1 - \xi, 2 + \xi\bigr]$ for $\xi = 4\sqrt{\log n / n}$.

2. Given that there are more than $4\sqrt{n \log n}$ elements that are at most $1/2$ in the original sequence s, the median is nevertheless larger than 2.

3. There is an interval of length $1/n$ that contains more than $\log n$ elements.

If any of these events happens, we bound the number of comparisons by the worst-case upper bound of $O(n^2)$. This contributes only $O(n)$ to the expected value, which is negligible. In the following, we assume that no bad event happens.

Let m denote the median. We assume from now on that $m \geq 1.5$. By symmetry (replacing s_i by $1 - s_i$ and ν_i by $2 - \nu_i$), the analysis for the the case $m \leq 1.5$ is identical.

We distinguish between *large* elements, which are larger than m, and *small* elements, which are smaller than m. To gain a better intuition, we take the following different view on the random process that generates $\overline{s}$. As before, we first generate $\overline{s}$ and then process it from left to right. In particular, this fixes the median m and it also fixes which elements are small and which elements are large. During this first process, we assume that none of the bad events 1, 2, and 3 happens.

In the second step, we redraw certain elements without changing the overall probability distribution: When a large pivot element $\overline{s}_i$ is encountered, we delete not only all elements larger than $\overline{s}_i$ (according to the algorithm), but we also redraw every large element $\overline{s}_j < \overline{s}_i$ uniformly at random from the interval $[m, \min\{\overline{s}_i, s_j + 2\}]$: Since $m \geq 1.5$, all these elements are eligible for $[m, s_j + 2]$ (A random number is eligible for an interval if it can take any value in this interval.) If $\overline{s}_i < s_j + 2$, we have to condition also on $\overline{s}_j \leq \overline{s}_i$.

Similarly, when a small pivot element $\overline{s}_i$ is encountered, we not only delete all elements smaller than $\overline{s}_i$, but also redraw every small element $\overline{s}_j > \overline{s}_i$ uniformly at random from the interval $[\max\{\overline{s}_i, s_j\}, \min\{m, s_j + 2\}]$: Any remaining small element is larger than $\overline{s}_i$. Furthermore, it is always at most $s_j + 2$ and, because it is small, also at most m. Redrawing the elements does not change the distribution of $\overline{s}$.

In fact, we do not actually have to redraw the elements, but we consider their distribution, conditioned on the fact that they assume a value in the given interval. The redrawing is only for intuition. Thus, we assume that none of the three bad events happens for $\overline{s}$ after redrawing certain elements.

We now argue that the number of pivot elements is in $O(\log n)$. Since every pivot element is compared to at most n other elements, this yields the desired bound of $O(n \log n)$ comparisons.

Note that a small element becomes a pivot element if and only if it is a left-to-right maximum among the sequence of small elements. Similarly, a large element is a pivot element if and only if it is a left-to-right *minimum* among the sequence of large elements. We determine the number of left-to-right minima and maxima separately.

We first deal with the number of pivot elements among the large elements. If at some point all large elements lie in an interval of length $1/n$, then we know that there are at most $O(\log n)$ large elements remaining. (Otherwise, we have bad event 3.) These elements can only contribute $O(n \log n)$ comparisons. We show that we only need a logarithmic number of iterations to ensure that all remaining large elements lie in such a small interval. So in total only a logarithmic number of large elements become a pivot element.

Lemma 3.8. *After processing $12 \log n$ large pivot elements, all remaining large elements lie in the interval $[m, m + \frac{1}{n}]$ with a probability of at least $1 - n^{-2}$.*

Proof. Let $\overline{s}_i^\ell$ denote the i-th large pivot element. Let $[m, c]$ denote the interval for which $\overline{s}_i^\ell$ is eligible. By construction, $\overline{s}_i^\ell$ is drawn uniformly at random from this interval. So with a probability of $1/2$, it lies in the first half of its interval, i.e., $\mathbb{P}(\overline{s}_i^\ell \in [m, \frac{m+c}{2}]) = 1/2$.

After processing at most $12 \log n$ large pivot elements, we will have encountered at least $2 \log n$ pivot elements that lie in the first half of their eligible interval with sufficiently high probability. In particular, let X be the number of pivot elements among the first $12 \log n$ large elements that lie in the first half of their interval. Then, by Chernoff's bound [11, Theorem 1.1],

$$\mathbb{P}(X < 2\log n) \le \exp\left(\frac{-2(4\log n)^2}{12\log n}\right) \le \exp\bigl(-2\log n\bigr) \le n^{-2}.$$

The length of the interval that contains all remaining large elements is shrinked by a factor of 2 by each of these at least $2\log n$ large pivot elements. Thus, the interval containing all large elements has a length of at most $\frac{3}{2^{2\log n}} = \frac{3}{n^2} \in o(1/n)$. □

Now we can complete the proof of Lemma 3.7. By Lemma 3.8, the case when the remaining interval of the large elements is larger than $1/n$ only contributes $O(1)$ comparisons to the expected number of comparisons.

What remains to be done is to bound the number of small pivot elements. The technical difficulty is that it can happen that not all elements are eligible for an interval $[c, m]$ for some c. But this is only the case for elements that are very small, i.e., when $s_i \le \xi \le 1/2$, and $m > 2$, because we assume that bad event 1 has not happened.

Let us first consider the case $m \le 2$. By the same line of reasoning as in the proof of Lemma 3.8, we need at most $O(\log n)$ small pivot elements until all small elements are in the interval $[m - \frac{1}{n}, m]$. There are only $O(\log n)$ elements in this interval (by the assumption that we do not have bad event 3), which contributes again $O(\log n)$ pivot elements.

To finish the proof, we consider small elements for the case $m > 2$. Again, after at most $O(\log n)$ small pivots, with sufficiently high probability, we have a small pivot larger than $2 - \frac{1}{n}$. The interval $[2 - \frac{1}{n}, 2]$ contains at most $O(\log n)$ elements, because bad event 3 has not happened. Overall, small elements smaller than 2 contribute at most $O(\log n)$ pivots.

Now we have to pay special attention to the small elements in the interval $[2, m] \subseteq \bigl[2, 2 + 4\sqrt{\log n/n}\bigr]$ that are not eligible for the whole interval $[2, m]$. (We have $m \le 2 + 4\sqrt{\log n/n}$ because otherwise we would have bad event 1.) The reason why we cannot apply the same argument for the remaining interval is that there might be small elements that are not eligible for the whole interval and so we cannot ensure that in each iteration the interval shrinks by a factor of 2. Intuitively, most small elements should indeed be eligible for the whole interval. As pointed out above, only elements s_i with $s_i \le \xi$ could possibly fail to be eligible for the whole interval. We have ruled out that there are more than $4\sqrt{n \log n}$ elements smaller than $1/2$ in the original sequence: If we had more such elements and the median were still $m > 2$, then we had bad event 2. The probability for such an element to assume a value in the interval $[2, m]$ is $O(\sqrt{\log n/n})$. Thus, in expectation, we have only $O(\sqrt{n \log n} \cdot \sqrt{\log n/n}) = O(\log n)$ such elements. Hence, they contribute only $O(n \log n)$ comparisons.

All the other small elements are eligible for the whole interval $[2, m]$, so, by the same line of reasoning as in Lemma 3.8, we conclude that after encountering $O(\log n)$ such pivot elements, the remaining interval is of size $1/n$. By the assumption that bad event 3 has not happened, such an interval only contains $O(\log n)$ elements, which completes the proof. □

3.4 Lower Bound for $d = 2$

In this section, we show that the upper bound of Section 3.3 for $d = 2$ is actually tight. The main idea behind the next result is as follows: We make sure that the median is close to 1 or close to 2. Otherwise, if the median is bounded away from 1 and 2, then a reasoning along the lines of Lemma 3.4 would yield a linear upper bound. We choose the sequence such that the median is roughly 2. For that, most elements are set to 1. Only the first few elements (few here means $n^{1/4}$) are set to 0. They yield $\Omega(\log n)$ left-to-right maxima, and all these become pivot elements. Each of these pivot elements contributes a linear number of comparisons.

Lemma 3.9. *There exists a family $(s^{(n)})_{n\in\mathbb{N}}$, where $s^{(n)}$ has a length of n, such that*

$$\mathbb{E}\big(\text{c-find}_2(s^{(n)}, \lceil n/2 \rceil)\big) \in \Omega\big(n \cdot \log n\big).$$

Proof. Consider the sequence

$$s = s^{(n)} = (\underbrace{0, 0, \dots, 0}_{n^{1/4}}, \underbrace{1, 1, \dots, 1}_{n - n^{1/4}}).$$

The probability that the first $n^{1/4}$ elements of $\overline{s}$ are at most $2 - n^{-1/4}$ is

$$\left(\frac{2 - n^{-1/4}}{2}\right)^{n^{1/4}} = \left(1 - \frac{1}{2n^{1/4}}\right)^{n^{1/4}} \geq \frac{1}{2}.$$

The probability that one particular element of the last $n - n^{1/4}$ elements is greater than $2 - n^{-1/4}$ is $\frac{1+n^{-1/4}}{2}$. Thus, for sufficiently large n, we expect to see

$$\frac{1 + n^{-1/4}}{2} \cdot \big(n - n^{1/4}\big) = \frac{n + n^{3/4} - n^{1/4} - 1}{2} \geq \frac{n}{2}$$

such elements. Since the number of elements larger than the first $n^{1/4}$ elements is binomially distributed, with constant probability, at least $n/2$ of the last $n - n^{1/4}$ elements of $\overline{s}$ are greater than all of the first $n^{1/4}$ elements of $\overline{s}$. Since these two events are independent from each other, both observations together imply that the following two properties hold with constant probability:

1. The median of $\overline{s}$ is among the last $n - n^{1/4}$ elements.

2. All left-to-right maxima of the first $n^{1/4}$ elements of $\overline{s}$ have to be compared to all elements greater than $2 - n^{-1/4}$, and there are at least $n/2$ such elements.

The number of left-to-right maxima of the first $n^{1/4}$ elements of $\overline{s}$ is expected to be $\ln(n^{1/4}) + O(1) \in \Theta(\log n)$ [4], which proves the lemma. □

4 Scan Maxima with Median-of-three Rule

The results in this section serve as a basis for the analysis of both quicksort and Hoare's find with the median-of-three rule. In order to analyze the number of scan maxima with the median-of-three rule, we analyze this number with the maximum- and minimum-of-two rules. The following lemma justifies this approach.

Lemma 4.1. *For every sequence s, we have*

$$\text{max2-scan}(s) \leq \text{med3-scan}(s) \leq \text{min2-scan}(s).$$

Proof. Let us focus on the first inequality. The proof of the second follows along the same lines.

Let $m = (m_1, m_2, \ldots)$ be the pivot elements according to the median-of-three rule, i.e., $m_1 = \text{median}(s_1, s_{\lceil n/2 \rceil}, s_n)$, m_2 is the median of the first, middle, and last element of the sequence containing all elements greater than m_1, and so on. Likewise, let $m' = (m'_1, m'_2, \ldots)$ be the pivot elements according to the maximum-of-two rule.

Now our aim is to prove that $m'_i \geq m_i$ for all i. Since we take scan maxima until all elements are removed, in particular the maximum of s must be an element in both sequences m and m'. Thus, m is at least as long as m', which proves the lemma.

The proof of $m'_i \geq m_i$ is by induction on i. The case $i = 1$ follows from $\max(s_1, s_n) \geq \text{median}(s_1, s_{\lceil n/2 \rceil}, s_n)$.

Now assume that s' and s'' are the sequences of elements that are greater than m_{i-1} and m'_{i-1}, respectively. Let ℓ and ℓ' be their lengths. By the induction hypothesis, $m_{i-1} \leq m'_{i-1}$. Thus, s'' is a subsequence of s'. The only elements that s' contains that are not part of s'' are the elements of value at most m'_{i-1}.

We have $m'_i = \max(s''_1, s''_{\ell'})$, and $m_i = \text{median}(s'_1, s'_{\lceil \ell/2 \rceil}, s'_\ell) \leq \max(s'_1, s'_\ell)$. Now either $s'_1 = s''_1$ or $s'_1 \leq m'_{i-1} < s''_1$. The same holds for s'_ℓ and $s''_{\ell'}$, which proves the lemma. □

The reason for considering max2-scan and min2-scan is that it is hard to keep track of where the middle element with the median-of-three rule lies: Depending on which element actually becomes the pivot and which elements are greater than the pivot, the new middle position can be far from the previous middle position.

Let us first prove a lower bound for the number of scan maxima.

Lemma 4.2. *There exists a sequence s such that for all $d \geq 1/n$, we have*

$$\mathbb{E}\big(\text{max2-scan}_d(s)\big) \in \Omega\left(\sqrt{\frac{n}{d}} + \log n\right).$$

Proof. For simplicity, we assume that n is even. Let $s = (\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n/2-1}{n}, \frac{1}{2}, \frac{1}{2}, \frac{n/2-1}{n}, \ldots, \frac{1}{n})$. Let

$$\Gamma_i = \{i+1, i+2, \ldots, i + 2\sqrt{nd}\} \cup \{n-i, n-i-1, \ldots, n-i-2\sqrt{nd}+1\}$$

be the set of the $2\sqrt{nd}$ indices following i plus the $2\sqrt{nd}$ indices preceding $n-i+1$. Note that s_{Γ_i} for $i \leq n/2 - 2\sqrt{nd}$ contains the corresponding values of the first and second half of s.

Let us estimate the probability that at least one element of Γ_i is a scan maximum. If for all i this probability is at least some positive constant, then we immediately obtain a lower bound of $\Omega(\sqrt{n/d})$ by linearity of expectation. To see this, we can consider Γ_i for $i = 2k\sqrt{nd}$ and $k \in [O(\sqrt{n/d}]$. These sets Γ_i are disjoint. (It then still remains to prove the $\Omega(\log n)$ lower bound.)

Assume that there exist indices $j < j'$ such that $\overline{s}_i < \min(\overline{s}_j, \overline{s}_{j'})$ for all $i < j$ and for all $i > j'$. Then at least one of them is a scan maximum.

Fix any $i \leq \frac{n}{2} - 2\sqrt{nd}$. Figure 1 shows Γ_i and illustrates the event whose probability we want to estimate now. Remember that ν_i denotes the additive noise at position i. Assume the following holds:

1. $\nu_{i+1}, \ldots, \nu_{i+\sqrt{nd}} \leq d - \sqrt{d/n}$.

2. $\nu_{n-i}, \nu_{n-i-1}, \ldots, \nu_{n-i-\sqrt{nd}+1} \leq d - \sqrt{d/n}$.

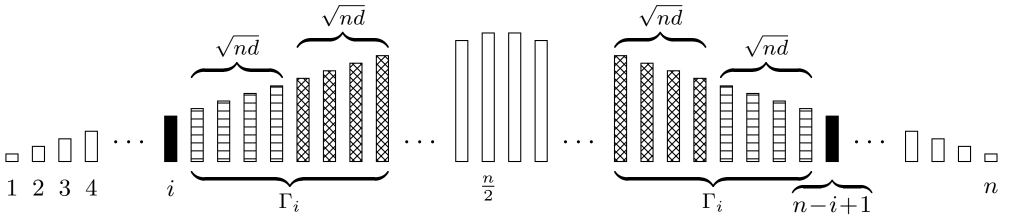

Figure 1: Γ_i consists of the $2\sqrt{nd}$ positions following position i and preceding the i-th last position, which is $n-i+1$. We estimate the probability that (1&2) none of the elements drawn with horizontal lines gets a huge noise added to it and (3) at least one of the elements drawn in crosshatch gets a huge noise and becomes a scan maximum.

3. There exist $j \in \Gamma_i$ such that $\nu_j > d - \sqrt{d/n}$.

Choose j to be minimal and j' to be maximal with property 3. Then by properties 1 and 2, $j > i+\sqrt{nd}$ and $j' \leq n-i-\sqrt{nd}$. If the three properties above are fulfilled, then, by the choice of j and j', $\overline{s}_j > \overline{s}_i$ for all $i < j$ and $i > j'$: For $i \in \Gamma_i$, this follows from the minimality of j and the maximality of j'. For $i \notin \Gamma$, $i \leq n/2$, we have $\overline{s}_i = \frac{i}{n} + \nu_i \leq \frac{i}{n} + d = \frac{i+\sqrt{nd}}{n} + d - \sqrt{d/n} \leq \overline{s}_j$ by the fact that $\nu_j > d - \sqrt{d/n}$. The case $i \notin \Gamma$, $i \geq n/2$ is similar. Thus, j or j' is a scan maximum.

Let us estimate the probability that this happens. We have

$$\mathbb{P}\left(\nu_{i+1}, \ldots, \nu_{i+\sqrt{nd}} \leq d - \sqrt{\frac{d}{n}}\right) = \left(\frac{d-\sqrt{d/n}}{d}\right)^{\sqrt{nd}} = \left(1 - \frac{1}{\sqrt{nd}}\right)^{\sqrt{nd}} \geq \frac{1}{4}$$

if $\sqrt{nd} \geq 2$. The latter is fulfilled if $d \geq 4/n$. If $d = c/n$ is smaller, we easily get a lower bound of $\Omega(n)$ by restricting the adversary to the interval $[0, c/4]$: We can apply the bound for $d = 4/n$ by scaling.

By symmetry, we also have

$$\mathbb{P}\left(\nu_{n-i}, \ldots, \nu_{n-i-\sqrt{nd}+1} \leq d - \sqrt{\frac{d}{n}}\right) \geq \frac{1}{4}.$$

Furthermore,

$$\mathbb{P}\left(\exists j \in \left\{i+\sqrt{nd}+1, \ldots, i+2\sqrt{nd}\right\} : \nu_j > d - \frac{d}{n}\right) = 1 - \left(\frac{d-\sqrt{d/n}}{d}\right)^{\sqrt{nd}} \geq 1 - \frac{1}{e}.$$

Overall, the probability that j and j' exist is constant, which proves the lower bound of $\Omega\bigl(\sqrt{n/d}\bigr)$.

To finish the proof, let us prove that, on average, we expect to see $\Omega(\log n)$ scan maxima. To do this, let us consider the sequence $s = (0, 0, \ldots, 0)$. We obtain $\overline{s}$ by adding noise from $[0, d]$. The ordering of the elements in $\overline{s}$ is now a uniformly distributed random permutation. We take a different view on the maximum-of-two pivot rule: We take s_1, get a half point for it and eliminate all elements smaller than s_1. If s_n has also been eliminated, then we have completed this iteration. Otherwise, we take s_n, get another half point and again eliminate all smaller elements. We repeat this procedure until the largest element is found.

The number of scan maxima of $\overline{s}$ is at least the number of points we get. Since the elements of $\overline{s}$ appear in random order, the expected number of points is $H_n/2$, where $H_n = \sum_{i=1}^{n} 1/i$ is the average-case number of scan maxima [4]. □

Now we turn to the upper bound for scan maxima.

Lemma 4.3. *For all sequences s and $d \geq 1/n$, we have*

$$\mathbb{E}\big(\text{min2-scan}_d(s)\big) \in O\left(\sqrt{\frac{n}{d}} + \log n\right).$$

Proof. First, we observe that a necessary condition for an element $\overline{s}_i$ to become a pivot element is that it is either a left-to-right maximum (according to the usual rule), i.e., no element $\overline{s}_j$ for $j < i$ is greater than $\overline{s}_i$, or that it is a right-to-left maximum, i.e., no element $\overline{s}_j$ for $j > i$ is greater than $\overline{s}_i$.

Hence, an upper bound for min2-scan($\overline{s}$) is the number of left-to-right maxima (c-scan($\overline{s}$)) plus the number of right-to-left maxima. The former is at most $O\big(\sqrt{n/d} + \log n\big)$ by (1), the latter can be analyzed in exactly the same way. Thus, the lemma follows. □

From Lemmas 4.1, 4.2, and 4.3 we immediately get tight bounds for the number of scan maxima with median-of-three rule.

Theorem 4.4. *For every $d \geq 1/n$, we have*

$$\max_{s \in [0,1]^n} \mathbb{E}\big(\text{med3-scan}_d(s)\big) \in \Theta\left(\sqrt{\frac{n}{d}} + \log n\right).$$

5 Quicksort and Hoare's Find with Median-of-three Rule

Now we use our results about scan maxima from the previous section to prove lower bounds for the number of comparisons that quicksort and Hoare's find need using the median-of-three pivot rule. We show that although in practice the median-of-three rule gives a better performance, it does not yield an asymptotically better bound. We only prove lower bounds here since they match already the upper bounds for the classic pivot rule. We strongly believe that the median-of-three rule does not yield worse bounds than the classic rule and, hence, that our bounds are tight. Our main goal of this section is to prove the following result for Hoare's find. This bound carries then over to quicksort.

Theorem 5.1. *For $d \geq 1/n$, we have*

$$\max_{s \in [0,1]^n} \mathbb{E}\big(\text{med3-find}_d(s)\big) \in \Omega\big(\tfrac{n}{d+1}\sqrt{n/d} + n\big).$$

Proof. We use the *maximum-of-two* rule to prove this lower bound. To this end, consider the following sequence: Let $\Delta = \{1, \ldots, \frac{n}{3}\} \cup \{\frac{2n}{3} + 1, \ldots, n\}$ and let s be defined by

$$s_i = \begin{cases} \min\big(\frac{i}{n}, \frac{n-1-i}{n}\big) & \text{if } i \in \Delta \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

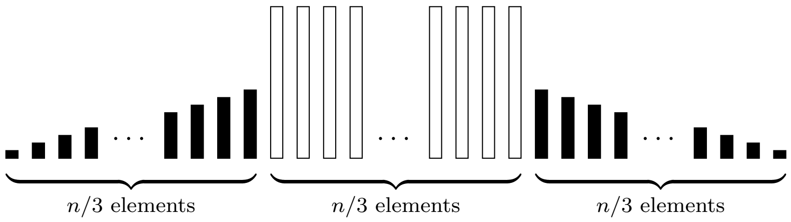

Figure 2: The sequence of Lemma 5.1. Black elements contribute scan maxima, white elements are large elements. All black scan maxima have to be compared to all or at least $\Omega(n/d)$ white elements.

Figure 2 gives an intuition how s looks like. We observe that s_Δ is, up to scaling, identical to the sequence used in Lemma 4.2. To analyze the number of comparisons, we distinguish between small and large values of d.

First, assume that $d \leq 2/3$. Then all elements of $\overline{s}_{[n]-\Delta}$ are greater than all elements of $\overline{s}_\Delta$, including the scan maxima of $\overline{s}_\Delta$. From Lemma 4.1 and the proof of Lemma 4.2, we know that $\overline{s}_\Delta$ contains $\Omega\big(\sqrt{n/d} + \log n\big)$ scan maxima. Each of these maxima has to be compared to all of the $n/3$ elements of $\overline{s}_{[n]-\Delta}$, resulting in $\Omega\big(n \cdot \big(\sqrt{n/d} + \log n\big)\big)$ comparisons.

The second case is $d \geq 2/3$. Again, there are $\Omega\big(\sqrt{n/d} + \log n\big)$ scan maxima under the maximum-of-two rule in $\overline{s}_\Delta$, which carry over to $\overline{s}$. According to Lemma 4.1, there are at least that many median-of-three scan maxima (med3 maxima) in $\overline{s}$, but since d may be greater than 2/3, some of the med3 maxima may be from $\overline{s}_{[n]\setminus\Delta}$. This causes no harm because the position of the pivots is of no relevance to the sorting process, but only their magnitude. In turn, the magnitude of a med3 maximum is at most the magnitude of the corresponding maximum-of-two scan maximum (max2 maximum).

We can now bound the number of comparisons appropriately. The probability that an element $\overline{s}_i$ ($i \in [n] \setminus \Delta$) is greater than the first $\Omega\big(\sqrt{n/d} + \log n\big)$ med3 scan maxima is at least the probability that it is greater than all maxima that are located in $\overline{s}_\Delta$, i.e.

$$\mathbb{P}\left(\overline{s}_i > \text{ first } \Omega\big(\sqrt{n/d} + \log n\big) \text{ med3 scan maxima}\right) \geq \mathbb{P}\left(1 + \nu_i > \frac{1}{3} + d\right) = \frac{2}{3d}.$$

Thus, by linearity of expectation, an expected number of $\Omega(n/d)$ elements of $\overline{s}_{[n]\setminus\Delta}$ are greater than the first $\Omega\big(\sqrt{n/d} + \log n\big)$ med3 scan maxima and have to be compared to all of them. This requires $\Omega\big(\frac{n}{d} \cdot \sqrt{n/d}\big)$ comparisons. Since we always need at least $\Omega(n)$ comparisons, the theorem follows. □

Since the number of comparisons that Hoare's find needs is a lower bound for the number of quicksort comparisons, we immediately get the following result for quicksort.

Corollary 5.2. *For $d \geq 1/n$, we have*

$$\max_{s \in [0,1]^n} \mathbb{E}\big(\text{med3-sort}_d(s)\big) \in \Theta\big(\tfrac{n}{d+1}\sqrt{n/d} + n \log n\big).$$

6 Hoare's Find Under Partial Permutations

To complement our findings about Hoare's find, we analyze the number of comparisons subject to partial permutations. For this model, we already have an upper bound of $O(\frac{n}{p} \log n)$ since the upper bound for quicksort [4] carries over to Hoare's find. We show that this is asymptotically tight (up to factors depending only on p) by proving that Hoare's find needs a smoothed number of $\Omega\big((1-p) \cdot \frac{n}{p} \cdot \log n\big)$ comparisons.

The main idea behind the proof of the following theorem is as follows: We aim at finding the median. The first few elements are close to and smaller than the median (few means roughly $\Theta\big((n/p)^{1/4}\big)$). Thus, it is unlikely that one of them is permuted further to the left. This implies that all unmarked of the first few elements become pivot elements. Then we observe that they have to be compared to many of the $\Omega(n)$ elements larger than the median, which yields our lower bound.

Theorem 6.1. *Let $p \in (0,1)$ be a constant. For every n there exists a sequence s of length n such that*

$$\mathbb{E}\big(\text{pp-find}_p(s)\big) \in \Omega\left((1-p) \cdot \frac{n}{p} \cdot \log n\right).$$

Proof. For simplicity, we restrict ourselves to odd n and permutations of $-m, -m+1, \ldots, m$ for $2m+1 = n$. This means that 0 is the median of the sequence. Let $Q = (m/p)^{1/4} \in \Theta\big((n/p)^{1/4}\big)$. We consider the sequence

$$s = (\underbrace{-Q, -Q+1, \ldots, -1}_{Q}, \underbrace{-m, \ldots, -Q-1}_{m-Q}, \underbrace{1, \ldots, m}_{m}, 0).$$

The important part of s is the first Q elements. All other elements can as well be in any other order.

Assume that the ith position is unmarked for some $i \leq Q$, i.e., $\overline{s}_i = s_i = -Q + i - 1$, and assume further that it becomes a pivot. The former happens with a probability of $1-p$. The latter means that all marked elements among $-Q+i, \ldots, -1$ are permuted further to the right (more precisely: not to the left of position i). To analyze how many comparisons $\overline{s}_i$ contributes, let

$$M_i = \min\big(\{\overline{s}_j \mid \overline{s}_j \geq 0, j < i\} \cup \{m+1\}\big).$$

Then $\overline{s}_i$ contributes at least M_i comparisons: All elements $0, \ldots, M_i - 1$ are to the right of position i. Thus, they are not already cut off by some other pivot. (In fact, $\overline{s}_i$ contributes at least $M_i + Q - i$ comparisons, but we ignore the $Q - i$ since it does not contribute to the asymptotics.) Let E^k be the event that the i-th position is unmarked, $\overline{s}_i = s_i$ becomes a pivot, and $M_i \geq k$. Using lower bounds for $\mathbb{P}\big(E^k\big)$, we get a lower bound for the expected number of comparisons.

From now on, we assume that $k \geq \sqrt{m/p}$. Let A be the number of marked positions prior to i, let B be the number of marked elements among $-Q+i, \ldots, -1$ and among $0, \ldots, k$, and let N be the total number of marked elements. We will see below that we can assume $A \leq B$. This allows us to estimate the probability of E^k as follows: We consider the B marked elements among $-Q+i, \ldots, -1, 0, \ldots, k$. The event E^k happens only if none of the elements is permuted to any of the marked positions prior to position i. If we consider these B elements one by one, the probability for the first not to assume such a position is $\frac{N-A}{N}$. For the second element, it is $\frac{N-A-1}{N-1}$, because we have already positioned the first element. This leaves us with only $N-1$ free positions overall and with only $N-A-1$ positions to the right of i, and so on. With this, we can bound the probability of E^k by

$$\begin{aligned}
W_k &= (1-p) \cdot \prod_{j=0}^{B-1} \frac{N-A-j}{N-j} \geq (1-p) \cdot \left(\frac{N-A-B}{N}\right)^A = (1-p) \cdot \left(1 - \frac{A+B}{N}\right)^A \\
&= (1-p) \cdot \exp\left(A \cdot \ln\left(1 - \frac{A+B}{N}\right)\right) \\
&\geq (1-p) \cdot \exp\left(-\frac{2A(A+B)}{N}\right) \geq (1-p) \cdot \exp\left(-\frac{4AB}{N}\right).
\end{aligned}$$

The first inequality holds since $A \leq B$ and hence most factors cancel out. The second inequality holds since $\ln(1-x) \geq -2x$ for $x \in [0, \frac{3}{4}]$. The third inequality again uses $A \leq B$.

This bound is monotonically decreasing in A and B, and monotonically increasing in N. Thus, we need upper bounds for A and B and a lower bound for N. Now let $1/p \leq i \leq Q - 1/p$, and let $k \geq \sqrt{m/p} = Q^2$. Assume that at most $2pi$ positions prior to i, at most $2p(Q-i)$ and at least $\frac{1}{2}p(Q-i)$ positions after i and before Q, at most $2pk$ and at least $\frac{1}{2}pk$ elements among $0, \ldots, k$ and at least $\frac{p}{2}n$ positions overall are marked. This yields $A \leq 2pi$, $A \leq B \leq 2pk + 2p(Q-i) \leq 3pk$ as well as $N \geq \frac{p}{2}n$. Since $i \geq 1/p$ and $Q - i \geq 1/p$, the probability that all these bounds are satisfied is at least a constant $c > 0$. This yields

$$W_k \geq c \cdot (1-p) \cdot \exp\left(-\frac{48pki}{n}\right) = c \cdot (1-p) \cdot K^k.$$

for $K = \exp(-48pi/n) \geq 1 - 48pi/n$. We observe that $K^{\sqrt{m/p}} \geq c' \in \Omega(1)$ and that K tends to 1 as n grows. Let X be the random number of comparisons with $\overline{s}_i$ as pivot element. Then, with the reasoning above, we have

$$\begin{aligned}
\mathbb{E}(X) &= \sum_{k=1}^{\infty} \mathbb{P}(X \geq k) \geq \sum_{k=1}^{m} \mathbb{P}(X \geq k) \geq \mathbb{P}\left(X \geq \sqrt{\frac{m}{p}}\right) \cdot \sqrt{\frac{m}{p}} + \sum_{k > \sqrt{m/p}} \mathbb{P}(X \geq k) \\
&\geq W_{\sqrt{m/p}} \cdot \sqrt{\frac{m}{p}} + \sum_{k > \sqrt{m/p}}^{m} W_k \\
&\geq cc' \cdot (1-p) \cdot \left(\sqrt{\frac{m}{p}} + \sum_{k=1}^{m-\sqrt{m/p}} K^k\right) \\
&\geq cc' \cdot (1-p) \cdot \sum_{k=1}^{m} K^k = cc' \cdot (1-p) \cdot K \cdot \frac{1-K^m}{1-K} \\
&\geq \frac{cc'}{2} \cdot (1-p) \cdot \frac{1}{1-K} \geq \frac{cc'}{2} \cdot (1-p) \cdot \frac{n}{96pi}.
\end{aligned}$$

The inequality in the third line holds exploits the lower bound for W_k derived above plus $K^{\sqrt{m/p}} \geq c'$. The inequality in the fourth line expoints $K^k \leq 1$ for all k, thus $\sqrt{m/p} \geq \sum_{k=m-\sqrt{m/p}+1}^{m} K^k$. The inequality in the fifth line exploits that $K \cdot (1 - K^{m+1}) \geq 1/2$ for sufficiently large n.

To finish the proof, we use linearity of expectation and sum over all $i \in \{1/p, \ldots, Q - 1/p\}$. This gives us the desired bound. □

To conclude this section and as a contrast to Sections 2 and 3, let us remark that for partial permutations, finding the maximum using Hoare's find seems to be easier than finding the median: The lower bound constructed above for finding the median requires that there are elements on either side of the element we aim for. If we aim at finding the maximum, all elements are on the same side of the target element. In fact, we believe that for finding the maximum and constant p, $O(n)$ comparisons suffice in expectation.

7 Concluding Remarks

We have shown tight bounds for the smoothed number of comparisons for Hoare's find under additive noise and under partial permutations. It turned out that, under additive noise, Hoare's find needs (asymptotically) more comparisons for finding the maximum than for finding the median. Furthermore, we analyzed quicksort and Hoare's find with the median-of-three pivot rule, and we proved that median-of-three does not yield an asymptotically better bound.

Our results for additive noise hold for arbitrary d. This includes large constants and even $d \in \omega(1)$. Such large values of d are of course mainly of theoretical interest. However, this is one of the few cases, where it is possible to obtain tight bounds in smoothed analysis.

A natural question regarding additive noise is what happens when the noise is drawn according to an arbitrary distribution rather than the uniform distribution. Some first results on this for left-to-right maxima were obtained by Damerow et al. [10]. We conjecture that uniform distributions are the worst case: If the adversary is allowed to specify a density function bounded by ϕ, then all upper bounds still hold with $d = 1/\phi$ (the maximum density of the uniform distribution on $[0, d]$ is $1/d$). However, a direct transfer of the results for uniform noise to arbitrary noise might be difficult [10].

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] David Arthur, Bodo Manthey, and Heiko Röglin. k-means has polynomial smoothed complexity. In *Proceedings of the 50th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 405–414. IEEE Computer Society, 2009.

[3] David Arthur and Sergei Vassilvitskii. Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method. *SIAM Journal on Computing*, 39(2):766–782, 2009.

[4] Cyril Banderier, René Beier, and Kurt Mehlhorn. Smoothed analysis of three combinatorial problems. In Branislav Rovan and Peter Vojtás, editors, *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 2747 of *Lecture Notes in Computer Science*, pages 198–207. Springer, 2003.

[5] Luca Becchetti, Stefano Leonardi, Alberto Marchetti-Spaccamela, Guido Schäfer, and Tjark Vredeveld. Average case and smoothed competitive analysis of the multilevel feedback algorithm. *Mathematics of Operations Research*, 31(1):85–108, 2006.

[6] René Beier, Heiko Röglin, and Berthold Vöcking. The smoothed number of Pareto optimal solutions in bicriteria integer optimization. In Matteo Fischetti and David P. Williamson, editors, *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 4513 of *Lecture Notes in Computer Science*, pages 53–67. Springer, 2007.

[7] René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences*, 69(3):306–329, 2004.

[8] René Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM Journal on Computing*, 35(4):855–881, 2006.

[9] Daniel Cederman and Philippas Tsigas. GPU-quicksort: A practical quicksort algorithm for graphics processors. *ACM Journal of Experimental Algorithms*, 14, 2009.

[10] Valentina Damerow, Bodo Manthey, Friedhelm Meyer auf der Heide, Harald Räcke, Christian Scheideler, Christian Sohler, and Till Tantau. Smoothed analysis of left-to-right maxima with applications. *ACM Transactions on Algorithms*, to appear.

[11] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press, 2009.

[12] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1295–1304. SIAM, 2007.

[13] Hannu Erkiö. The worst case permutation for median-of-three quicksort. *The Computer Journal*, 27(3):276–277, 1984.

[14] Mahmoud Fouz, Manfred Kufleitner, Bodo Manthey, and Nima Zeini Jahromi. Smoothed analysis of quicksort and Hoare's find. In Hung Q. Ngo, editor, *Proceedings of the 15th Annual International Computing and Combinatorics Conference (COCOON)*, volume 5609 of *Lecture Notes in Computer Science*, pages 158–167. Springer, 2009.

[15] C. A. R. Hoare. Algorithm 64: Quicksort. *Communications of the ACM*, 4(7):322, 1961.

[16] C. A. R. Hoare. Algorithm 65: Find. *Communications of the ACM*, 4(7):321–322, 1961.

[17] Peter Kirschenhofer and Helmut Prodinger. Comparisons in Hoare's find algorithm. *Combinatorics, Probability and Computing*, 7(1):111–120, 1998.

[18] Peter Kirschenhofer, Helmut Prodinger, and Conrado Martinez. Analysis of Hoare's find algorithm with median-of-three partition. *Random Structures and Algorithms*, 10(1-2):143–156, 1997.

[19] Donald E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming.* Addison-Wesley, 2nd edition, 1998.

[20] Bodo Manthey and Rüdiger Reischuk. Smoothed analysis of binary search trees. *Theoretical Computer Science*, 378(3):292–315, 2007.

[21] Bodo Manthey and Heiko Röglin. Improved smoothed analysis of k-means clustering. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 461–470. SIAM, 2009.

[22] Bodo Manthey and Heiko Röglin. Worst-case and smoothed analysis of k-means clustering with Bregman divergences. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Proceedings of the 20th Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 5878 of *Lecture Notes in Computer Science*, pages 1024–1033. Springer, 2009.

[23] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[24] Ralph Neininger. *Limit Laws for Random Recursive Structures and Algorithms.* PhD thesis, Universität Freiburg, 1999.

[25] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007.

[26] Douglas C. Schmidt. `qsort.c`. C standard library `stdlib` within `glibc 2.7`, available at `http://ftp.gnu.org/gnu/glibc/`, 2007.

[27] Robert Sedgewick. The analysis of quicksort programs. *Acta Informatica*, 7(4):327–355, 1977.

[28] Robert Sedgewick. Implementing quicksort programs. *Communications of the ACM*, 21(10):847–857, 1978.

[29] Richard C. Singleton. Algorithm 347: An efficient algorithm for sorting with minimal storage. *Communications of the ACM*, 12(3):185–186, 1969.

[30] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[31] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.