Approximate Selection with Unreliable Comparisons in Optimal Expected Time

Huang, Shengyu; Liu, Chih Hung; Rutschmann, Daniel

Published in:
Proceedings of the 40th International Symposium on Theoretical Aspects of Computer Science

Link to article, DOI:
10.4230/LIPIcs.STACS.2023.37

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Approximate Selection with Unreliable Comparisons in Optimal Expected Time

Shengyu Huang
Department of Computer Science, EPFL, Lausanne, Switzerland

Chih-Hung Liu
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Daniel Rutschmann
Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Copenhagen, Denmark

Abstract

Given n elements, an integer $k \leq \frac{n}{2}$ and a parameter $\varepsilon \geq \frac{1}{n}$, we study the problem of selecting an element with rank in $(k - n\varepsilon, k + n\varepsilon]$ using *unreliable* comparisons where the outcome of each comparison is incorrect independently with a constant error probability, and multiple comparisons between the same pair of elements are independent. In this fault model, the fundamental problems of finding the minimum, selecting the k-th smallest element and sorting have been shown to require $\Theta\big(n \log \frac{1}{Q}\big)$, $\Theta\big(n \log \frac{k}{Q}\big)$ and $\Theta\big(n \log \frac{n}{Q}\big)$ comparisons, respectively, to achieve success probability $1 - Q$ [9]. Considering the increasing complexity of modern computing, it is of great interest to develop approximation algorithms that enable a trade-off between the solution quality and the number of comparisons. In particular, approximation algorithms would even be able to attain a sublinear number of comparisons. Very recently, Leucci and Liu [23] proved that the approximate minimum selection problem, which covers the case that $k \leq n\varepsilon$, requires expected $\Theta(\varepsilon^{-1} \log \frac{1}{Q})$ comparisons, but the general case, i.e., for $n\varepsilon < k \leq \frac{n}{2}$, is still open.

We develop a randomized algorithm that performs *expected* $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ comparisons to achieve success probability at least $1 - Q$. For $k = n\varepsilon$, the number of comparisons is $O(\varepsilon^{-1} \log \frac{1}{Q})$, matching Leucci and Liu's result [23], whereas for $k = n/2$ (i.e., approximating the median), the number of comparisons is $O(\varepsilon^{-2} \log \frac{1}{Q})$. We also prove that even in the absence of comparison faults, any randomized algorithm with success probability at least $1 - Q$ performs *expected* $\Omega\big(\min\{n, \frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q}\}\big)$ comparisons. As long as n is large enough, i.e., when $n = \Omega(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$, our lower bound demonstrates the optimality of our algorithm, which covers the possible range of attaining a sublinear number of comparisons. Surprisingly, for constant Q, our algorithm performs expected $O(\frac{k}{n}\varepsilon^{-2})$ comparisons, matching the best possible approximation algorithm *in the absence of computation faults*. In contrast, for the exact selection problem, the expected number of comparisons is $\Theta(n \log k)$ with faults versus $\Theta(n)$ without faults. Our results also indicate a clear distinction between approximating the minimum and approximating the k-th smallest element, which holds even for the high probability guarantee, e.g., if $k = \frac{n}{2}$, $Q = \frac{1}{n}$ and $\varepsilon = n^{-\alpha}$ for $\alpha \in (0, \frac{1}{2})$, the asymptotic difference is almost quadratic, i.e., $\tilde{\Theta}(n^{\alpha})$ versus $\tilde{\Theta}(n^{2\alpha})$.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Approximate Selection, Unreliable Comparisons, Independent Faults

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.37

Related Version *Full Version*: `https://arxiv.org/abs/2205.01448` [17]

Funding *Chih-Hung Liu*: Yushan Young Fellow Program by Ministry of Education, Taiwan and Research Project 111-2222-E-002-017-MY2 by National Science and Technology Council, Taiwan.

Acknowledgements The three authors began to investigate this topic when all of them were in ETH Zürich, Switzerland.

1 Introduction

We study a generalization of the *k-th smallest element selection* problem in terms of *approximation* and *fault tolerance*. Given a set S of n elements, an integer k and a parameter ε, the *fault-tolerant ε-approximate k-selection* problem, FT-APX(k, ε) for short, is to return an element with rank in $(k - n\varepsilon, k + n\varepsilon]$ only using *unreliable* comparisons whose outcome can be *incorrect*. Due to the comparison faults, it is impossible to guarantee a correct solution, so the number of comparisons performed by an algorithm should depend on the *failure probability* Q of the algorithm where $Q < \frac{1}{2}$. We assume that $k \le \frac{n}{2}$ and $\varepsilon \ge \frac{1}{n}$; if $k > \frac{n}{2}$, the problem becomes to approximate the $(n-k)$-th largest element, which is symmetric, and if $\varepsilon < \frac{1}{n}$, the problem becomes the exact selection problem. The elements with rank in $(0, k - n\varepsilon]$, $(k - n\varepsilon, k + n\varepsilon]$ and $(k + n\varepsilon, n]$ of S are called *small*, *relevant* and *large*, respectively.

We consider *independent random comparison faults*: There is a strict ordering relation among S, but algorithms can only gather information via unreliable comparisons between two elements. The outcome of each comparison is wrong with a known constant probability $\boldsymbol{p} < \frac{1}{2}$. When comparing the same pair of elements multiple times, each outcome is *independent* of the previous outcomes; comparisons involving different pairs of elements are also independent.

The above fault model has been widely studied for various fundamental problems such as finding the minimum, selecting the k-th smallest (resp. largest) element and sorting a sequence [9, 29, 30]. Feige et al [9] proved that to achieve success probability $1 - Q$, the aforementioned three problems require $\Theta\big(n \log \frac{1}{Q}\big)$, $\Theta\big(n \log \frac{k}{Q}\big)$ and $\Theta\big(n \log \frac{n}{Q}\big)$ comparisons, respectively, both in *expectation* and in the *worst case*. In the sequel, their selection algorithm is denoted by Select(k, Q), and its performance is summarized as follows.

▶ **Theorem 1** ([9]). Select(k, Q) *performs* $O\big(n \log \frac{k}{Q}\big)$ *comparisons to select the k-th smallest (resp. largest) element among n elements with success probability at least* $1 - Q$.

Due to the increasing complexity of modern computing, error detection and error correction require enormous computing resources. Emerging technologies enable the tolerance of computation errors for saving computing resources [28, 16, 7, 19, 32]. Meanwhile, many practical applications do not require an optimal answer, but just a good enough one. These circumstances motivate the study of fault-tolerant approximation algorithms, especially for the possibility of attaining a sublinear number of comparisons.

Recently, Leucci and Liu [23] studied the approximate minimum selection problem, which asks for one element with rank in $[1, n\varepsilon]$ and thus is equivalent to FT-APX(k, ε) with $k = 0$ (since FT-APX(k, ε) seeks one element with rank in $(k - n\varepsilon, k + n\varepsilon]$ under our formulation). They developed an algorithm using *expected* $O(\varepsilon^{-1} \log \frac{1}{Q})$ comparisons and also proved a matching lower bound. Moreover, if $k \le n\epsilon$, a correct answer to FT-APX$(0, \varepsilon)$ is also correct to FT-APX(k, ε), indicating that this case is essentially the approximate minimum selection. As a result, the challenge is to tackle the case that $n\epsilon < k \le \frac{n}{2}$.

A straightforward approach to the FT-APX(k, ε) problem is to first *randomly* pick $m = \Theta(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ elements so that the underlying $\lceil m \cdot \frac{k}{n} \rceil$-th smallest element is *relevant* with probability at least $1 - \frac{Q}{2}$, and then apply Select$(\lceil m \cdot \frac{k}{n} \rceil, \frac{Q}{2})$ on the m elements. By Theorem 1, this approach requires $\Theta\big(\frac{k}{n}\varepsilon^{-2}\big((\log \frac{1}{Q})(\log \frac{k}{n}\varepsilon^{-2}) + \log^2 \frac{1}{Q}\big)\big)$ comparisons.

▶ Remark 2. Using standard sampling techniques, e.g., similar to Leucci and Liu's ideas [23], the number of comparisons in the above straightforward approach can easily be improved to $O\big(\frac{k}{n}\varepsilon^{-2}(\log \frac{1}{Q})(\log \frac{k}{n}\varepsilon^{-2}) + \log^2 \frac{1}{Q}\big)$. However, for constant Q, this number remains $O\big(\frac{k}{n}\varepsilon^{-2}(\log \frac{k}{n}\varepsilon^{-2})\big)$, and there is no obvious way of improving it to $O(\frac{k}{n}\varepsilon^{-2})$. Remark 8 in Section 4 will discuss how variants of Quickselect run into a similar issue.

Table 1 Summary for the known results and our new results.

	Minimum	k-th Element
Exact	$\Theta(n \log \frac{1}{Q})$ [9]	$\Theta(n \log \frac{k}{Q})$ [9]
Approximate	$\Theta(\varepsilon^{-1} \log \frac{1}{Q})$ [23]	$O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ [ours] $\Omega\big(\min\{n, \frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q}\}\big)$ [ours]
Exact without faults	$\Theta(n)$	$\Theta(n)$
Approximate without faults	$\Theta\big(\min\{n, \varepsilon^{-1} \log \frac{1}{Q}\}\big)$	$\Theta\big(\min\{n, \frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q}\}\big)$ [ours]

To sum up, it is of great interest to study if the FT-APX(k, ε) problem can be solved with probability $1 - Q$ using $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ comparisons. Moreover, since comparison faults increase the number of comparisons required for the exact selection problem, i.e., from $\Theta(n)$ without faults to $\Theta(n \log \frac{k}{Q})$ with faults, which shows a clear gap even for constant Q, one may wonder if the same phenomenon occurs for the approximate selection problem. Furthermore, although finding the minimum and finding the k-th smallest element require different numbers of comparisons, namely $\Theta(n \log \frac{1}{Q})$ versus $\Theta(n \log \frac{k}{Q})$, to attain a high success probability, i.e., $Q = \frac{1}{n}$, both problems require $\Theta(n \log n)$ comparisons. Hence, it is also desirable to investigate if there is a stronger distinction between these two problems in the approximation scenario, especially for the high success probability.

1.1 Our Contributions

We develop a randomized algorithm that performs *expected* $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ comparisons to solve the FT-APX(k, ε) problem with probability at least $1 - Q$. Also, we prove that even without considering comparison faults, any randomized algorithm with success probability $1 - Q$ requires *expected* $\Omega\big(\min\{n, \frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q}\}\big)$ comparisons. As long as n is large enough, i.e., when $n = \Omega(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$, our lower bound demonstrates the optimality of our algorithm, which covers the possible range of attaining a sublinear number of comparisons. Table 1 summarize the known results and our new results.

Furthermore, for any constant Q, e.g., $Q = 1/4$, our algorithm performs expected $O(\frac{k}{n}\varepsilon^{-2})$ comparisons, which is optimal even in the absence of comparison faults. This surprising outcome distinguishes the approximate selection problem from the exact selection problem, where the exact selection problem requires expected $\Theta(n)$ comparisons without faults, but expected $\Theta(n \log k)$ comparisons with faults.

Moreover, our results also indicate that there is a distinction between the approximate *minimum* selection problem and the general approximate *k-th element* selection problem in terms of the expected number of comparisons, i.e., $\Theta(\varepsilon^{-1} \log \frac{1}{Q})$ [24] versus $\Theta(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$. This distinction even holds for the high probability guarantee ($Q = \frac{1}{n}$) in contradiction to the fact that the two problems have the same complexity $\Theta(n \log n)$ in the *exact* selection (Theorem 1). For example, if $k = \frac{n}{2}$ and $Q = \frac{1}{n}$, the two approximate selection problems require expected $\Theta(\varepsilon^{-1} \log n)$ and $\Theta(\varepsilon^{-2} \log n)$ comparisons, respectively. If $\varepsilon = n^{-\alpha}$ for a constant $\alpha \in (0, \frac{1}{2})$, the asymptotic difference is almost quadratic, i.e., $\tilde{\Theta}(n^{\alpha})$ versus $\tilde{\Theta}(n^{2\alpha})$.

▶ Remark 3. The $\frac{k}{n}\varepsilon^{-2}$ term in the above complexities is actually $\max\{\varepsilon^{-1}, \frac{k}{n}\varepsilon^{-2}\}$. As discussed before, the case that $k \le n\epsilon$, by which $\varepsilon^{-1} \ge \frac{k}{n}\varepsilon^{-2}$, belongs to the approximate minimum selection problem. Therefore, to simplifying the description, we assume that $k > n\epsilon$ throughout the paper if no further specification.

As noted in Remark 2, our technical advance is to improve the $\frac{k}{n}\varepsilon^{-2}(\log\frac{1}{Q})(\log\frac{k}{n}\varepsilon^{-2}) + \log^2\frac{1}{Q}$ term to $\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q}$. To some extent, compared with Leucci and Liu's algorithms, our algorithm covers the entire range of k instead of the case when k is trivially small. In addition, our algorithm owns an elegant feature that it only exploits simple sampling techniques, e.g., selecting the median of three samples and selecting the minimum of two samples.

The top-level of our algorithms, inspired by Leucci and Liu [23], reduces the FT-APX(k, ε) problem on n elements to the FT-APX$(\frac{m}{2}, \frac{3}{8})$ problem on $m = \Theta(\log\frac{1}{Q})$ elements. More precisely, if a relevant element can be selected with probability $\frac{8}{9}$, we can generate a sequence of $\Theta(\log\frac{1}{Q})$ elements in which $\frac{3}{4}$ of elements around the middle, with probability $1 - \frac{Q}{2}$, are all relevant. For such a "dense" sequence, we design a delicate trial-and-error method to select a relevant element with probability $1 - \frac{Q}{2}$ using expected $\Theta(\log\frac{1}{Q})$ comparisons.

The main challenge is to obtain a relevant element with probability $\frac{8}{9}$ using only $O(\frac{k}{n}\varepsilon^{-2})$ comparisons. For the approximate minimum ($k = 0$), Leucci and Liu [23] applied Select$(1, \frac{1}{10})$ on $\Theta(\varepsilon^{-1})$ randomly picked elements and attained $O(\varepsilon^{-1})$ comparisons. However, for general k, this method requires $\Theta(\frac{k}{n}\varepsilon^{-2}\log\frac{k}{n}\varepsilon^{-2})$ comparisons with an extra logarithmic factor.

We first work on a special case that $k = \frac{n}{2}$, i.e., the approximate median selection. Based on the symmetry property of the median, we observe that the median of three randomly picked elements is more likely to be relevant than a randomly picked element. We exploit this observation to iteratively increase the ratio of relevant elements while keeping the underlying median being relevant. Once the ratio becomes a constant fraction, we will apply a straightforward method.

For general k, we design a "purifying" process that iteratively increases the ratio of relevant elements while keeping elements around a "controlled" position being relevant. Despite no symmetry property, we still observe that under certain conditions, the minimum of two randomly picked elements is more likely to be relevant than a randomly picked one. Then, we derive feasible parameters to control the relative position of k, i.e., the middle of the remaining relevant elements, during the purifying process. Once the relative position becomes a constant fraction of the remaining elements, we add dummy smallest elements and apply our approximate median selection.

As a by-product, we also give a randomized algorithm using *deterministic* $O\big(\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q} + (\log\frac{1}{Q})(\log\log\frac{1}{Q})^2\big)$ comparisons, and it is still open how to attain deterministic $O\big(\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q}\big)$ comparisons. Besides, when $\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q} = \omega(n)$, we derive another lower bound of $\Omega(\max\{n, \varepsilon^{-1}\log\frac{(k+n\varepsilon)/(2n\varepsilon)}{Q}\})$ (Theorem 21). In this situation, a trivial upper bound of $O(n\log\frac{k}{Q})$ follows from Theorem 1. It is also not clear how to fill this gap between the lower and the upper bounds, i.e., $\Omega(\max\{n, \varepsilon^{-1}\log\frac{(k+n\varepsilon)/(2n\varepsilon)}{Q}\})$ versus $O(n\log\frac{k}{Q})$.

The rest of the paper is organized as follows. Section 1.2 gives a brief literature review. Section 2 provides a few preliminary remarks. Section 3 presents the top-level algorithm. Section 4 and Section 5 describe sub-algorithms to approximate the median and the k-th element with constant probability, respectively. Section 6 sketches the lower bound analysis. Appendices A–D include several technical details omitted from the main text. For other technical details not included in this manuscript, interested readers are referred to the current full version [17].

1.2 Brief Literature

Dating back to the 1987, Ravikumar et al. [31] already studied a variant of the problem of finding the *exact* minimum using unreliable comparisons when at most f comparison faults are allowed. They proved that $\Theta(fn)$ comparisons are necessary in the worst case. Later, Aigner [1] considered a *prefix-bounded* error model: for a fraction parameter $\gamma < \frac{1}{2}$, at most an γ-fraction of the past comparisons failed at any point during the execution of an algorithm. He proved that $\Theta(\frac{1}{1-p})^n$ comparisons are necessary to find the minimum in the worst case. Furthermore, he proved that if $p > \frac{1}{n-1}$, no algorithm can succeed with certainty [1].

When errors occur independently, as already discussed, Feige et al. [9] showed that the number of comparisons required for selecting the exact k-th smallest element with success probability at least $1 - Q$ is $\Theta(n \log \frac{k}{Q})$. Recently, Braverman et al. [4] investigated the *round complexity* and the number of comparisons by partition and selection algorithms. They proved that for any constant error probability, $\Theta(n \log n)$ comparisons are necessary for any algorithm that selects the minimum with high probability. Also, Chen et al. [6] studied the problem of computing the smallest k elements using r given independent noisy comparisons between each pair of elements. In a very general error model called *strong stochastic model*, they gave a linear-time algorithm with competitive ratio of $\tilde{O}(\sqrt{n})$, and also proved that this competitive ratio is tight.

The related problem of sorting with faults has also received considerable attention. When there are at most f comparison faults, $\Theta(n \log n + fn)$ comparisons are necessary and sufficient to correctly sort n elements [21, 25, 2]. For the prefix-bounded model, although Aigner's result on the minimum selection [1] implies that $(\frac{1}{1-p})^{O(n \log n)}$ are sufficient to sort n elements, Borgstrom and Kosaraju [3] showed that checking whether the input elements are sorted already requires $\Omega\big((\frac{1}{1-p})^n\big)$ comparisons. When comparison faults are permanent, or equivalently, when a pair of elements can only be compared once, the underlying sorting problem has also been extensively studied especially because it can be connected to both the *minimum feedback arc set* problem and the *rank aggregation* problem [26, 18, 4, 5, 20, 22, 14, 11, 13, 12]. There are also sorting algorithms for memory faults [10, 24]. For more knowledge about fault-tolerant search algorithms, we refer the interested readers to a survey by Pelc [30] and a monograph by Cicalese [8].

2 Preliminary

As explained in the beginning of Section 1 and in Remark 3, we assume that $n\epsilon < k \leq \frac{n}{2}$ and $\epsilon \geq \frac{1}{n}$ throughout the paper if no further specification. For ease of exposition, we use $\boldsymbol{\beta}$ to denote $\frac{\boldsymbol{k}}{\boldsymbol{n}}$ in some analyses, and we sometimes abuse the name x of an element to denote its rank, e.g., we might write "$x \in [l, r]$" to denote that the rank of x lies in the range $[l, r]$. Comparing two elements, x and y, yields an outcome of either $x < y$ or $y > x$. A typical subroutine in our algorithms is to draw elements using sampling with replacement, so multiple copies of an element may appear in a set. When two copies of the same element are compared, the tie is broken using any arbitrary (but consistent) ordering among the copies.

In our fault model, there is a standard strategy called *majority vote* for reducing the "error probability" of comparing two elements. We state this strategy as follows.

▶ **Lemma 4** (Majority Vote). *For any error probability $p \in [0, \frac{1}{2})$, there exists a postive integer c_p such that a strategy that compares two elements $2c_p \cdot t + 1$ times and returns the majority result succeeds with probability at least $1 - e^{-t}$, where $c_p = \lceil \frac{4(1-p)}{(1-2p)^2} \rceil$. The exact failure probability of this strategy is*

$$\sum_{i=0}^{c_p \cdot t} \binom{2c_p \cdot t + 1}{i} (1-p)^i p^{2c_p \cdot t + 1 - i}.$$

Many analyses in this paper will make use of the following Chernoff bound.

▶ **Lemma 5** (Chernoff Bound). *Let X be the sum of independent Bernoulli random variables. If $A \le E[X] \le B$, then for any $\delta \in (0,1)$,*

$$\Pr[X \ge (1+\delta)\cdot B] \le e^{-\frac{\delta^2}{3}B} \qquad \text{and} \qquad \Pr[X \le (1-\delta)\cdot A] \le e^{-\frac{\delta^2}{2}A}.$$

3 Top Level of Algorithm

The high-level idea is to reduce solving FT-APX(k,ε) on n elements with probability at least $1-Q$ to solving FT-APX$(\frac{m}{2},\frac{3}{8})$ on $m=\Theta(\log\frac{1}{Q})$ elements with probability at least $1-\frac{Q}{2}$. Specifically, if a *relevant* element can be selected with probability at least $\frac{8}{9}$, then m selected elements, for some $m=\Theta(\log\frac{1}{Q})$, contain at least $\frac{7}{8}m$ relevant elements with probability at least $1-\frac{Q}{2}$; see Lemma 23 in Appendix B. In this situation, at least $2\cdot(\frac{7}{8}-\frac{1}{2})\cdot m = 2\cdot\frac{3}{8}m$ elements around the median, i.e., the range $(\frac{1}{8}m,\frac{7}{8}m]$, are relevant. Therefore, solving the FT-APX$(\frac{m}{2},\frac{3}{8})$ problem on these m elements with probability at least $1-\frac{Q}{2}$ yields a relevant element with probability at least $1-2\cdot\frac{Q}{2}=1-Q$.

Section 5 will present an approach that uses $O(\frac{k}{n}\varepsilon^{-2})$ comparisons to select a relevant element with probability at least $\frac{8}{9}$, by which the above reduction takes $O(\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q})$ comparisons. In the remaining of this section, we will explain how to solve FT-APX$(\frac{m}{2},\frac{3}{8})$ with probability $1-\frac{Q}{2}$ efficiently in both expectation and determination.

We first design a simple trial-and-error method that uses *expected* $O(\log\frac{1}{Q})$ comparisons to select an element from $(\frac{1}{8}m,\frac{7}{8}m]$ with probability at least $1-\frac{Q}{2}$:

> Repeatedly pick an element randomly and verify if its rank lies in $(\frac{1}{8}m,\frac{7}{8}m]$ until one element passes the verification.

Since $(\frac{1}{8}m,\frac{7}{8}m]$ contains $\frac{3}{4}m$ elements, the expected number of repetitions before encountering a correct element is only $O(1)$. Therefore, the key is to implement the verification step such that the method returns a correct element with probability at least $1-\frac{Q}{2}$ and the expected number of comparisons is $O(\log\frac{1}{Q})$.

We implement the *verification step* for an element x based on a simple experiment that randomly picks three elements, and checks if x is neither the smallest nor the largest among the four elements. To simply the follow-up analysis, we assume that the three elements are sampled with replacement and picking x again is allowed. Under the above assumptions, the probability that the if-condition holds is $1-(\frac{r_x}{m})^3-(1-\frac{r_x}{m})^3=\frac{3}{4}-3(\frac{r_x}{m}-\frac{1}{2})^2$ where r_x is the rank of x among the m elements, $(\frac{r_x}{m})^3$ is the probability that none of the three picked element is larger than x and $(1-\frac{r_x}{m})^3$ is the probability that none of the three picked element is smaller than x. Also, the check can be conducted with success probability at least $\frac{17}{18}$ using $O(1)$ comparisons (by plugging in $n=4$, $k=1$ and $Q=\frac{1}{36}$ into Theorem 1 twice for the smallest and largest versions, respectively). Therefore, if $x\in(\frac{2}{8}m,\frac{6}{8}m]$, the experiment succeeds with probability *at least* $\left(\frac{3}{4}-3(\frac{2}{8}-\frac{1}{2})^2\right)\cdot\frac{17}{18}=\frac{17}{32}$, where $\left(\frac{3}{4}-3(\frac{2}{8}-\frac{1}{2})^2\right)$ is the minimum probability that the if-condition holds and $\frac{17}{18}$ is the success probability of the check, while if $x\in[1,\frac{1}{8}m]$ or $x\in(\frac{7}{8}m,m]$, the experiment succeeds with probability *at most* $\left(\frac{3}{4}-3(\frac{1}{8}-\frac{1}{2})^2\right)+\frac{1}{18}=\frac{221}{576}\le\frac{15}{32}$, where $\left(\frac{3}{4}-3(\frac{1}{8}-\frac{1}{2})^2\right)$ is the maximum probability that the if-condition holds and $\frac{1}{18}$ is the failure probability of the check.

In the above derivation, we ignore two ranges $(\frac{1}{8}m,\frac{2}{8}m]$ and $(\frac{6}{8}m,\frac{7}{8}m]$ since all elements in these two ranges are relevant, by which returning any element in these two ranges will not decrease the success probability, and since the considered range $(\frac{2}{8}m,\frac{6}{8}m]$ contains enough elements. Based on the above calculated probabilities, we can conceptually treat the above

simple experiment as an unreliable comparison with error probability $\frac{15}{32}$. By Lemma 4, if the verification step conducts this simple experiment $2 \cdot c_{15/32} \ln \frac{2}{Q} + 1$ times and takes the majority result, its success probability is at least $1 - \frac{Q}{2}$.

Now, we are ready to analyze the expected number of comparisons and the success probability of our trial-and-error method. Roughly speaking, the process of this trial-and-error method is similar to flipping a coin until a head appears where the probability of getting a head is at least $\frac{1}{4}$, which corresponds to a geometric distribution, and each flip requires $O(\log \frac{1}{Q})$ comparisons. More precisely, a single round returns an element in $(\frac{2}{8}m, \frac{6}{8}m]$ with probability *at least* $\frac{1}{2} \cdot (1 - \frac{Q}{2}) \geq \frac{1}{4}$, and thus the probability to conduct the i-th round is at most $(\frac{3}{4})^{i-1}$. Therefore, the expected number of comparisons is at most $\sum_{i \geq 1} (\frac{3}{4})^{i-1} \cdot (2 \cdot c_{15/32} \ln \frac{2}{Q} + 1) = O(\log \frac{1}{Q})$. Besides, this method fails only if it returns an element in $[1, \frac{1}{8}m]$ or $(\frac{7}{8}m, m]$, and such probability of a single round is *at most* $\frac{1}{4} \cdot \frac{Q}{2} = \frac{Q}{8}$. Again, since the probability to conduct the i-th round is at most $(\frac{3}{4})^{i-1}$, the failure probability is at most $\sum_{i \geq 1} (\frac{3}{4})^{i-1} \cdot \frac{Q}{8} = \frac{Q}{2}$, concluding the following theorem:

▶ **Theorem 6.** *It takes* ***expected*** $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ *comparisons to solve the FT-APX*(k, ε) *problem with probability at least* $1 - Q$.

Finally, to derive a deterministic bound, we note that the simple experiment in the verification step may be viewed as a *biased* coin toss. From this viewpoint, we are able to turn the FT-APX$(\frac{m}{2}, \frac{3}{8})$ problem into finding a coin with bias bigger than $\frac{15}{32}$, given that at least half of the coins have bias at least $\frac{17}{32}$. Grossman and Moshkovitz [15] provided an algorithm that solves the new problem with probability $1 - \frac{Q}{2}$ using $O(\log \frac{1}{Q} \cdot (\log \log \frac{1}{Q})^2)$ coin tosses, leading to the following theorem.

▶ **Theorem 7.** *It takes* $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q} + \log \frac{1}{Q} (\log \log \frac{1}{Q})^2)$ *comparisons to solve the FT-APX*(k, ε) *problem with probability at least* $1 - Q$.

4 Approximate Median Selection

We attempt to select an element in $(\frac{n}{2} - n\varepsilon, \frac{n}{2} + n\varepsilon]$, i.e., $\boldsymbol{k = \frac{n}{2}}$, with probability at least $1 - \frac{1}{18}$ using only $O(\varepsilon^{-2})$ comparisons. This algorithm will then be applied in Section 5 as a subroutine. A straightforward method, denoted by ST-Median(ε), picks $m = \Theta(\varepsilon^{-2})$ elements randomly to make their median *relevant* with probability at least $1 - \frac{1}{72}$ and applies the Select$(\frac{m}{2}, \frac{1}{72})$ algorithm (Theorem 1), resulting in a failure probability of at most $\frac{1}{36}$. However, the Select$(\frac{m}{2}, \frac{1}{72})$ algorithm takes $O(m \log m) = O(\varepsilon^{-2} \log \varepsilon^{-1})$ comparisons with an *extra* logarithmic factor. To achieve $O(\varepsilon^{-2})$ comparisons, we will "purify" the input elements in a way that the ratio of relevant elements is increasing while the underlying median is still relevant. Once the ratio of relevant elements becomes a constant fraction, i.e., from 2ε to $\Omega(1)$, we can afford to apply the ST-Median algorithm. We assume that $\varepsilon < \frac{1}{6}$ since if $\varepsilon \geq \frac{1}{6}$, the ST-Median$(\varepsilon)$ algorithm takes only $O(\varepsilon^{-2} \log \varepsilon^{-1}) = O(1)$ comparisons.

▶ Remark 8. A major difficulty to overcome in the purifying process is the following: if we consider three elements that are each relevant with probability ρ, then their median, even in the absence of comparison faults, is relevant with probability at most $\frac{3}{2}\rho + O(\rho^2)$, which is a lot less than 3ρ. Thus, we risk running out of elements long before the ratio of relevant elements becomes a constant. This issue remains if we replace three by a larger constant, and it applies to any algorithm that works in a non-constant number of phases, including algorithms that more closely resemble Quickselect. Those algorithms would need to start with $\Omega(\varepsilon^{-(2+\delta)})$ elements for some $\delta > 0$ and hence cannot achieve the $O(\varepsilon^{-2})$ bound.

To settle the issue in Remark 8, we maintain a multiset of elements and re-sample from this multiset at every phase. Our re-sampling method allows us to decrease the number of elements by less than a factor of $\frac{3}{2}$, so we can avoid running out of elements.

The algorithm is sketched as follows:

1. For $1 \le i \le L$, generate a multiset M_i of n_i elements by repeatedly picking three elements from M_{i-1} *randomly* and selecting the median of the three using a symmetric median selection algorithm (Lemma 9 below).
2. Apply the ST-Median($\boldsymbol{\varepsilon_L}$) algorithm on M_L.

Initially, $M_0 = S$, $n_0 = n$, $\varepsilon_0 = \varepsilon$. M_i is called ***good*** if all elements in the range $(\frac{n_i}{2} - n_i\varepsilon_i, \frac{n_i}{2} + n_i\varepsilon_i]$ are *relevant*. Moreover, n_i is decreasing with i while ε_i is increasing with i, and $L = \min\{i \mid \varepsilon_i \ge \frac{1}{6}\}$, i.e., the minimum of number of rounds such that at least $2 \cdot \frac{1}{6} = \frac{1}{3}$ of the elements around the middle is relevant. The rest of this section illustrates the idea behind this process and implements these parameters n_i and ε_i.

▶ **Lemma 9.** *For **three** elements, consider the following median selection algorithm:*

1. *For each pair of elements, apply the majority vote strategy with $2c_p \cdot 4 + 1$ comparisons (Lemma 4), and assign a point to the element that attains the majority result.*
2. *Return the element with exactly one point. If all three elements get exactly one point, return one of them uniformly at random.*

The above algorithm returns the median with probability at least $1 - \frac{1}{13}$, and returns the minimum and the maximum with the same probability, i.e., at most $\frac{1}{26}$.

The purifying process is inspired by a simple observation: a randomly picked element is relevant with probability 2ε, while the *median* of three randomly picked elements is relevant with probability much greater than 2ε. Let E_S denote the event that the median of three randomly picked elements is small. Then,

$$\Pr[E_S] = 3\left(\frac{1}{2} - \varepsilon\right)^2\left(\frac{1}{2} + \varepsilon\right) + \left(\frac{1}{2} - \varepsilon\right)^3 = \frac{1}{2} - \frac{3}{2}\varepsilon + 2\varepsilon^3.$$

If $\varepsilon < \frac{1}{6}$, then $\Pr[E_S] \le \frac{1}{2} - \frac{3}{2}\varepsilon + 2(\frac{1}{6})^2\varepsilon = 1 - \frac{13}{9}\varepsilon$. By Lemma 9, the median selection returns the median with probability at least $1 - \frac{1}{13}$, and returns the minimum (resp. the maximum) with probability at most $\frac{1}{26}$. A simple calculation, together with the above arguments, gives the following lemma:

▶ **Lemma 10.** *If M_{i-1} is good, then each element in M_i is small (resp. large) with probability at most $\frac{1}{2} - \frac{4}{3}\varepsilon_{i-1}$.*

Proof. We only prove the small case, and it is symmetric to the large case. Let p_s denote the probability that an element randomly picked from M_{i-1} is small. Since M_{i-1} is good, all elements in its range $(\frac{1}{2}n_{i-1} - n_{i-1}\varepsilon_{i-1}, \frac{1}{2}n_{i-1} + n_{i-1}\varepsilon_{i-1}]$ are relevant, and $p_s \le \frac{1}{2} - \varepsilon_{i-1}$. Let p_1, p_2 and p_3 denote the probabilities that the median selection algorithm in Lemma 9 returns the minimum, the median and the maximum of three elements, respectively. By Lemma 9, $p_2 \ge \frac{12}{13}$, and $p_1 = p_3 \le \frac{1}{26}$. Also recall that $\varepsilon_{i-1} \le \frac{1}{6}$. Then, the probability that an element in M_i is small is

$$
\begin{aligned}
&\underbrace{p_s^3}_{\text{three small}} + \underbrace{3p_s^2(1-p_s)}_{\text{two small \& one non-small}} \cdot(1-p_3) + \underbrace{3p_s(1-p_s)^2}_{\text{one small \& two non-small}} \cdot p_1 \\
&= p_s^3 + 3p_s^2(1-p_s)(1-p_1) + 3p_s(1-p_s)^2 \cdot p_1 \\
&= \left(3p_s^2 - 2p_s^3\right) + p_1 \cdot \underbrace{\left(3p_s - 9p_s^2 + 6p_s^3\right)}_{\geq 0 \text{ since } 0 \leq p_s \leq \frac{1}{2}} \\
&\overset{p_1 \leq \frac{1}{26}}{\leq} \frac{1}{26} \cdot \underbrace{\left(3p_s + 69p_s^2 - 46p_s^3\right)}_{f(x) := -46x^3+69x^2+3x \ \&\ f'(x)>0 \text{ for } 0 \leq x \leq 1} \\
&\overset{p_s \leq \frac{1}{2} - \varepsilon_{i-1}}{\leq} \frac{1}{26} \cdot \left(3\left(\frac{1}{2} - \varepsilon_{i-1}\right) + 69\left(\frac{1}{2} - \varepsilon_{i-1}\right)^2 - 46\left(\frac{1}{2} - \varepsilon_{i-1}\right)^3\right) \\
&= \frac{1}{2} - \frac{75}{52}\varepsilon_{i-1} + \frac{23}{13}\varepsilon_{i-1}^3 \\
&\overset{\varepsilon_{i-1} < \frac{1}{6}}{\leq} \frac{1}{2} - \frac{75}{52}\varepsilon_{i-1} + \frac{23}{468}\varepsilon_{i-1} \\
&= \frac{1}{2} - \frac{652}{468}\varepsilon_{i-1} \\
&\leq \frac{1}{2} - \frac{4}{3}\varepsilon_{i-1}
\end{aligned}
$$

◀

By Lemma 10, it is feasible to set $\varepsilon_i = (\frac{5}{4})^i \cdot \varepsilon$, i.e., growing slightly slower than $\frac{4}{3}$. Then, the size n_i is set as $\lceil 2000 \cdot i \cdot (\frac{4}{5})^{\mathbf{2i}} \cdot \varepsilon^{-2} \rceil$ to limit the number of comparisons and the failure probability. First, n_i is linear in ε^{-2} since the minimum number of elements to be looked at is $\Omega(\varepsilon^{-2})$ (Section 6). Second, to bound the total number of comparisons, n_i should shrink exponentially with i. Third, to bound the failure probability of the algorithm, the failure probability of the i-th round should also shrink exponentially with i. From the above three aspects, since the Chernoff bound (Lemma 5) will be applied for the probabilistic analysis, n_i should be linear in i, and the shrink factor of n_i should be at least $(\frac{4}{5})^2$ to cancel out the square of the growth factor $\frac{5}{4}$ of ε_i.

Because the ST-Median(ε_L) algorithm (stated at the beginning of this section) fails with probability at most $\frac{1}{36}$, it is sufficient to prove that $\Pr[M_L \text{ is good }] \geq 1 - \frac{1}{36}$. Let E_i denote the event that M_i is good. By definition, $\Pr[E_0] = 1$. With the Chernoff bound, we can prove the following lemma:

▶ **Lemma 11.** *For $1 \leq i \leq L$,*

$$\Pr[M_i \textit{ is NOT good } \mid M_{i-1} \textit{ is good }] \leq 2 \cdot e^{-5i}.$$

Proof. Assume that M_{i-1} is good. Let X_i be the number of small elements in M_i and let Y_i be the number of large elements in M_i. For the statement, it is sufficient to prove that $\Pr[X_i \geq \frac{n_i}{2} - n_i\varepsilon_i] \leq e^{-5i}$ and $\Pr[Y_i \geq \frac{n_i}{2} - n_i\varepsilon_i] \leq e^{-5i}$. We will prove the first claim, and it is symmetric to the second claim. By Lemma 10,

$$E[X_i] \leq (\frac{1}{2} - \frac{4}{3}\varepsilon_{i-1})n_i = (\frac{1}{2} - \frac{4}{3} \cdot \frac{4}{5}\varepsilon_i)n_i = (\frac{1}{2} - \frac{16}{15}\varepsilon_i)n_i = \frac{15 - 32\varepsilon_i}{30}n_i.$$

By Lemma 5 (Chernoff bound), we can get

$$\begin{aligned}
\Pr\left[X_i \geq \left(\frac{1}{2}-\varepsilon_i\right)n_i\right] &= \Pr\left[\left(1+\frac{\frac{1}{15}\varepsilon_i}{\frac{1}{2}-\frac{16}{15}\varepsilon_i}\right)\left(\frac{1}{2}-\frac{16}{15}\varepsilon_i\right)n_i\right] \\
&= \Pr\left[\left(1+\underbrace{\frac{2\varepsilon_i}{15-32\varepsilon_i}}_{:=\delta}\right)\underbrace{\left(\frac{15-32\varepsilon_i}{30}n_i\right)}_{\geq E[X_i]}\right] \\
&\underbrace{\leq \exp\left(-\frac{1}{3}\left(\frac{2\varepsilon_i}{15-32\varepsilon_i}\right)^2\cdot\left(\frac{15-32\varepsilon_i}{30}n_i\right)\right)}_{\text{Lemma 5}} \\
&= \exp\left(-\frac{4}{90}\cdot\frac{\varepsilon_i^2}{15-32\varepsilon_i}\cdot n_i\right) \leq \exp\left(-\frac{4}{90}\cdot\frac{\varepsilon_i^2}{15}\cdot n_i\right) \\
&= \exp\left(-\frac{2}{675}\cdot\left(\left(\frac{5}{4}\right)^{2i}\varepsilon^2\right)\cdot\left(2000\cdot i\cdot\left(\frac{4}{5}\right)^{2i}\cdot\varepsilon^{-2}\right)\right) \\
&\leq e^{-5i}.
\end{aligned}$$

◀

By Lemma 11, we can lower bound $\Pr[E_L]$ as

$$\Pr[E_L] = 1-\Pr[\bigcup_{i=1}^{L}\overline{E_i}\mid E_{i-1}] \geq 1-\sum_{i=1}^{L}2\cdot e^{-5i} \geq 1-4\cdot e^{-5} \geq 1-\frac{1}{36}.$$

By Lemma 9, each median selection takes $O(1)$ comparisons, so the purifying process takes $O(\sum_{i=1}^{L}n_i) = O\big(\varepsilon^{-2}\sum_{i=1}^{L}i\cdot(\frac{4}{5})^{2i}\big) = O(\varepsilon^{-2})$ comparisons. Since $\varepsilon_L \geq \frac{1}{6}$, the ST-Median($\varepsilon_L$) algorithm takes $O(1)$ comparisons, concluding the following theorem:

▶ **Theorem 12.** *It takes $O(\varepsilon^{-2})$ comparisons to select an element in $(\frac{n}{2}-n\varepsilon, \frac{n}{2}+n\varepsilon]$ with probability at least $1-\frac{1}{18}$.*

5 Approximate k-th Element Selection

We attempt to select an element in $(k-n\varepsilon, k+n\varepsilon]$ with probability at least $1-\frac{1}{9}$ using only $O(\frac{k}{n}\varepsilon^{-2})$ comparisons. Recall that $k > n\varepsilon$ as assumed in Remark 3. If $n\varepsilon < k \leq 2n\varepsilon$, we halve the value of ε so that $k > 2n\varepsilon$, which does not increase the asymptotic complexity. Therefore, we can safely assume $k > 2n\varepsilon$ afterwards. In this scenario, the straightforward approach mentioned in Section 1 requires $O\big(\frac{k}{n}\varepsilon^{-2}\log(k\varepsilon^{-1})\big)$ comparisons with an extra $\log(k\varepsilon^{-1})$ factor. Another approach is to add $n-2k$ dummy smallest elements (so that the relevant elements lie in the middle) and to apply the algorithm in Section 4 with $\frac{\varepsilon}{2}$, leading to $O(\varepsilon^{-2})$ comparisons. As a result, both approaches are more expensive than $O(\frac{k}{n}\varepsilon^{-2})$.

At a high level, our breakthrough is an iterative "purifying" process that increases both the ratio of relevant elements and the relative position of k, i.e., the middle position of relevant elements, while "controlling" the relative position. Once the relative position becomes a constant fraction of the remaining elements, e.g., $\frac{1}{8}$, we add dummy smallest elements and apply the approximate median selection algorithm in Section 4. As the ratio of relevant elements increases at the same time, the resulting number of comparisons will be $O(\frac{k}{n}\varepsilon^{-2})$ instead of $O(\varepsilon^{-2})$.

The algorithm is sketched as follows:

1. For $1 \leq i \leq L$, generate a set S_i of n_i elements by repeatedly picking two elements from S_{i-1} *randomly* and selecting the minimum of the two using $2\mathrm{c}_p\cdot 3+1$ comparisons (Lemma 4).

2. Add $n_L - 2k_L$ dummy smallest elements to M_L and apply the approximate median selection algorithm in Section 4 on M_L with respect to ε_L.

Initially, $S_0 = S$, $n_0 = n$, $k_0 = k$, $\varepsilon_0 = \varepsilon$. S_i is called ***good*** if all elements in the range $(k_i - n_i\varepsilon_i,\, k_i + n_i\varepsilon_i]$ are *relevant*. For ease of exposition, let β_i denote $\frac{k_i}{n_i}$. Both β_i and ε_i increase with i while n_i decreases with i. We set $L = \min\{i \mid \beta_i \geq \frac{1}{8}\}$. Recall that $\beta = \frac{k}{n}$. We assume that $\beta < \frac{1}{8}$; otherwise, we conduct the second step directly, i.e., $L = 0$.

The purifying process is based on a simple observation that the minimum of two randomly picked element is *small* with probability

$$\underbrace{(\beta-\varepsilon)^2}_{\text{two small}} + \underbrace{2\,(\beta-\varepsilon)\,(1-(\beta-\varepsilon))}_{\text{one small \& one non-small}} = 2\,(\beta-\varepsilon) - (\beta-\varepsilon)^2\,,$$

while a randomly picked element is small with probability merely $\beta - \varepsilon$. By a similar calculation, the minimum of two randomly picked elements is *relevant* with $4\varepsilon - \beta \cdot 4\varepsilon$. Since k is exactly the number of small elements plus half the number of relevant elements, the above derivation suggests the following formulation of β_i:

$$\beta_i := \underbrace{2\,(\beta_{i-1}-\varepsilon_{i-1}) - (\beta_{i-1}-\varepsilon_{i-1})^2}_{\Pr[\text{ small }]} + \underbrace{(2\varepsilon_{i-1} - \beta_{i-1}\cdot 2\varepsilon)}_{\Pr[\text{ relevant }]\div 2}\,.$$

These derivations need to adapt to the failure probability q of selecting the minimum using $2c_p \cdot 3 + 1$ comparisons. By Lemma 4, $q \leq e^{-3} < \frac{1}{20}$ and $q = \sum_{i=1}^{3c_p}(1-p)^i p^{6c_p+1-i}$. Then, a selected element in the first round is *relevant* with probability

$$\underbrace{4\varepsilon^2}_{\text{two relevant}} + q \cdot \underbrace{2\cdot(\beta-\varepsilon)\,2\varepsilon}_{\text{one small \& one relevant}} + (1-q)\cdot \underbrace{2\cdot(1-(\beta+\varepsilon))\,2\varepsilon}_{\text{one large \& one relevant}}\,,$$

which is equal to $4\varepsilon \cdot \big((1-q) - (1-2q)\cdot\beta\big)$. Since $\beta < \frac{1}{8}$ and $q < \frac{1}{20}$, the above probability is larger than $\frac{67}{40}\cdot 2\varepsilon$. Therefore, it is feasible to set $\boldsymbol{\varepsilon_i = (\frac{3}{2})^i \cdot \varepsilon}$, i.e., growing slower than $\frac{67}{20}$.

To fit the formulation of β_i to the above failure probability q, a similar calculation yields that each selected element in the first round is *small* with probability

$$(\beta-\varepsilon)^2 + (1-q)\cdot 2\,(\beta-\varepsilon)\,(1-(\beta-\varepsilon))\,.$$

Since the relative position is the number of small elements plus half the number of relevant elements, it is feasible to set the value of β_i as follows (after arrangement):

$$\beta_i := \left(2\beta_{i-1} - \beta_{i-1}^2 - \varepsilon_{i-1}^2\right) - 2q\left(\beta_{i-1} - \beta_{i-1}^2 - \varepsilon_{i-1}^2\right).$$

Moreover, we can prove by induction important properties of β_i as stated below:

▶ **Lemma 13.** *For* $0 \leq i \leq L$,

$$\beta_i > 2\varepsilon_i \quad \textit{and} \quad \beta_i \leq 2^i\cdot\beta. \quad \textit{Thus,} \quad \frac{k_i}{n_i} \leq 2^i \cdot \frac{k}{n} \quad \textit{for} \quad 0 \leq i \leq L.$$

Proof. We prove by induction. For $i = 0$, by assumption in the first paragraph of Section 5, $\beta > 2\varepsilon$, i.e., $\beta_0 = \beta > 2\varepsilon = 2\varepsilon_0$. Also, $\beta_0 = \beta \leq 2^0 \cdot \beta$. Assume that for $i = k \geq 0$, $\beta_k > 2\varepsilon_k$ and $\beta_k \leq 2^k \cdot \beta$. Note that $k < L$; otherwise, the $(k+1)$-th round does not exist. By Section 5,

$$\beta_{k+1} = \left(2\beta_k - \beta_k^2 - \varepsilon_k^2\right) - 2q\left(\beta_k - \beta_k^2 - \varepsilon_k^2\right).$$

We first prove that $\beta_{k+1} > 2\varepsilon_{k+1}$ as follows:

$$\begin{aligned}
\beta_{k+1} &= \left(2\beta_k - \beta_k^2 - \varepsilon_k^2\right) - 2q\left(\beta_k - \beta_k^2 - \varepsilon_k^2\right) \overset{q<\frac{1}{20}}{>} \frac{19}{10}\beta_k - \frac{9}{10}(\beta_k^2 + \varepsilon_k^2) \\
&= \frac{9}{10}\left(-\left(\beta_k - \frac{19}{18}\right)^2 + \left(\frac{19}{18}\right)^2 - \varepsilon_k^2\right) \\
&\overset{\beta_k<\frac{1}{8}\ \&\ 2\varepsilon_k<\beta_k}{>} \frac{9}{10}\left(-\left(2\varepsilon_k - \frac{19}{18}\right)^2 + \left(\frac{19}{18}\right)^2 - \varepsilon_k^2\right) \\
&= \frac{19}{5}\varepsilon_k - \frac{9}{2}\varepsilon_k^2 \overset{\varepsilon_k<\frac{1}{2}\beta_k<\frac{1}{16}}{>} \frac{563}{160}\varepsilon_k > 3\varepsilon_k = 2\varepsilon_{k+1}.
\end{aligned}$$

Then, we prove that $\beta_{k+1} \le 2^{k+1} \cdot \beta$ as follows:

$$\begin{aligned}
\beta_{k+1} &= \left(2\beta_k - \beta_k^2 - \varepsilon_k^2\right) - 2q\left(\beta_k - \beta_k^2 - \varepsilon_k^2\right) \overset{q\ge 0}{\le} 2\beta_k - \beta_k^2 - \varepsilon_k^2 \\
&\le 2\beta_k \le 2 \cdot 2^k \cdot \beta = 2^{k+1} \cdot \beta.
\end{aligned}$$

◀

The size n_i of S_i is set as $\lceil 960 \cdot i \cdot (\frac{8}{9})^i \cdot \frac{k}{n}\varepsilon^{-2} \rceil$ to control the number of comparisons and the failure probability. Similar to Section 4, n_i should shrink exponentially with i and should also be linear in both $\frac{k}{n}\varepsilon^{-2}$ and i. The major difference lies in that the existence of k_i changes the shrink factor of n_i. Since $\frac{k_i}{n_i} \le 2^i \cdot \frac{k}{n}$ (by Lemma 13) and $\varepsilon_i = (\frac{3}{2})^i \cdot \varepsilon$, the shrink factor of n_i should be at least $\frac{8}{9}$. This is based on the fact that $2^{-i} \cdot (\frac{3}{2})^{2i} \cdot (\frac{8}{9})^i = 1$, which will be much clearer in the probability analysis.

To sum up, $\boldsymbol{n_i} = \lceil 960 \cdot i \cdot (\frac{8}{9})^i \cdot \frac{k}{n}\varepsilon^{-2} \rceil$, $\boldsymbol{\varepsilon_i} = (\frac{3}{2})^i \cdot \varepsilon$, $\boldsymbol{\beta_i} = (2\beta_{i-1} - \beta_{i-1}^2 - \varepsilon_{i-1}^2) - 2q \cdot (\beta_{i-1} - \beta_{i-1}^2 - \varepsilon_{i-1}^2)$ with $\boldsymbol{q} = \sum_{i=1}^{3\mathrm{c}_p}(1-p)^i p^{6\cdot \mathrm{c}_p + 1 - i}$, and $\boldsymbol{L} = \min\{i \mid \beta_i \ge \frac{1}{8}\}$.

To attain the success probability $1 - \frac{1}{9}$, since the approximate median selection in Section 4 fails with probability at most $\frac{1}{18}$, it is sufficient to prove that $\Pr[S_L \text{ is good}] \ge 1 - \frac{1}{18}$ (Theorem 12). Let E_i denote the event that S_i is good. By definition, $\Pr[E_0] = 1$. Applying the Chernoff bound with the above parameters gives the following lemma: (The proof is rather technical, and interested readers are referred to the current full version [17].)

▶ **Lemma 14.** *For* $1 \le i \le L$

$$\Pr[S_i \textit{ is NOT good} \mid S_{i-1} \textit{ is good}] \le 2 \cdot e^{-4.3 \cdot i}.$$

By Lemma 14, we can lower bound $\Pr[E_L]$ as

$$\Pr[E_L] = 1 - \Pr[\bigcup_{i=1}^{L} \overline{E_i} \mid E_{i-1}] \ge 1 - \sum_{i=1}^{L} 2 \cdot e^{-4.3 \cdot i} \ge 1 - 4 \cdot e^{-4.3} \ge 1 - \frac{1}{18}.$$

For the number of comparisons, since each selection takes $2\mathrm{c}_P \cdot 3 + 1 = O(1)$ comparisons, the purifying process takes $\sum_{i=1}^{L} O(n_i) = \frac{k}{n}\varepsilon^{-2} \cdot \sum_{i=1}^{L} O\big(i \cdot (\frac{8}{9})^i\big) = O(\frac{k}{n}\varepsilon^{-2})$ comparisons. By Theorem 12, the approximate median selection takes $O(\varepsilon_L^{-2}) = O\big((\frac{2}{3})^{2L} \cdot \varepsilon^{-2}\big) = O(2^{-L} \cdot \varepsilon^{-2})$ comparisons. Since $\frac{k_L}{n_L} \le 2^L \cdot \frac{k}{n}$ (Lemma 13) and $\frac{k_L}{n_L} = \beta_L \ge \frac{1}{8}$, we have $2^{-L} = O(\frac{k}{n})$ and $O(2^{-L} \cdot \varepsilon^{-2}) = O(\frac{k}{n}\varepsilon^{-2})$, implying the following main theorem:

▶ **Theorem 15.** *It takes* $O(\frac{k}{n}\varepsilon^{-2})$ *comparisons to select an element in* $(k - n\varepsilon, k + n\varepsilon]$ *with probability at least* $1 - \frac{1}{9}$.

6 Lower Bound

We sketch the derivation of an $\Omega(\min\{n, \frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q}\})$ lower bound for the *expected* number of comparisons. Our derivation contains two key ingredients. First, we design an auxiliary decision tree that simulates any corresponding randomized algorithm with success probability at least $1-Q$, but owns nice properties for the analysis. Second, we derive a sampling lemma (Corollary 18) that lower bounds the probability of a returned element being relevant.

We assume that $\boldsymbol{4n\varepsilon \le k}$. If $k \le n\varepsilon$, the $\Omega(\varepsilon^{-1}\log\frac{1}{Q})$ lower bound for the approximate minimum selection problem [23] applies, and if $n\varepsilon < k < 4n\varepsilon$, we multiply the value of ε by 4 so that $k \le n\varepsilon$ and the former argument still works, which does not change the lower bound asymptotically. We assume that there are **no comparison faults**, which does not increase the lower bound and is easier for analysis.

Let T be the decision tree of any *randomized* algorithm that solves FT-APX(k,ε) with probability at least $1-Q$. T is said to *look at* an element x if T performs at least one comparison involving x. Let $\mathfrak{D}$ be the *expected* number of elements that T looks at. Since $\mathfrak{D}$ is not larger than twice the expected number of comparisons, it is sufficient to lower bound $\mathfrak{D}$. If $\mathfrak{D} \ge \frac{n}{10}$, then $\mathfrak{D} = \Omega(n)$. Below, we deal with the case that $\mathfrak{D} < \frac{n}{10}$.

We construct an auxiliary decision tree $\tilde{T}$ based on T: $\tilde{T}$ first simulates T until reaching a leaf u of T that returns an element x, and then conducts three additional steps *sequentially*:

(a) If T does not look at x, then $\tilde{T}$ compares x with another element.

(b) If $\tilde{T}$ has looked at fewer than $2\mathfrak{D} + \lceil\frac{8n}{k}\rceil$ elements so far, then $\tilde{T}$ performs more comparisons such that $\tilde{T}$ has looked at *exactly* $2\mathfrak{D} + \lceil\frac{8n}{k}\rceil$ elements after this step.

(c) $\tilde{T}$ compares all pairs of elements that it has looked at, and then returns x.

Intuitively, $\tilde{T}$ represents the same algorithm as T, but these additional steps will give $\tilde{T}$ nice properties for analysis. Roughly speaking, Step (a) ensures that $\tilde{T}$ must look at the returned element. The term $2\mathfrak{D} + \lceil\frac{8n}{k}\rceil$ in Step (b) comes from that by Markov's inequality, T looks at more than $2\mathfrak{D}$ elements with probability at most $\frac{1}{2}$, and that the $\lceil\frac{8n}{k}\rceil$ term will cancel out an $\frac{k}{n}$ term later. Step (c) enables $\tilde{T}$ to know the sorted order of the elements that $\tilde{T}$ looked at. These three steps lead to three nice properties in the following lemma.

▶ **Lemma 16.** *$\tilde{T}$ has the following properties:*

(1) *$\tilde{T}$ knows the sorted order of the elements that $\tilde{T}$ has looked at.*

(2) *$\tilde{T}$ has success probability at least $1-Q$.*

(3) *$\tilde{T}$ looks at exactly $2\mathfrak{D} + \lceil\frac{8n}{k}\rceil$ elements with probability at least $1/2$. Note that this includes the elements that $\tilde{T}$ looks at during its simulation of T.*

Proof. Property (1) comes from step (c) in which $\tilde{T}$ compares all pairs of elements that $\tilde{T}$ has looked at. Remember that we assume no comparison faults for the lower bound analysis.

For property (2), note that $\tilde{T}$ first simulates T, then does some additional comparison and then returns the element that T would have returned (independent of the outcome of the additional comparisons). Hence $\tilde{T}$ has the same success probability as T, which is at least $1-Q$ by assumption.

For property (3), according to the three steps, if T looks at no more than $2\mathfrak{D}$ elements, then $\tilde{T}$ will look exactly $2\mathfrak{D} + \lceil\frac{8n}{k}\rceil$ elements. Since the probability that T looks at more than $2\mathfrak{D}$ elements is at most $\frac{1}{2}$ (by the definition of $\mathfrak{D}$ and by Markov's inequality), property (3) follows. ◀

Let us consider the execution of $\tilde{T}$ on a uniformly shuffled input. Recall that we assume there are no comparison faults. According to Lemma 16(1), the element returned by a fixed leaf of $\tilde{T}$ will always have the same rank among the elements that $\tilde{T}$ has looked at,

independent of the order of the input. Under this circumstance, it is desirable to analyze the relevance of elements with a certain rank among all sampled elements. Toward this end, we derive a sampling lemma (Lemma 17) that lower bounds the probability of an element with a small sample rank being small and the probability of an element with a large sample rank being large, which further induces a key sampling lemma (Corollary 18) that lower bounds the probability of an element being relevant.

Corollary 18 roughly states that for a set A of randomly sampled elements (without replacement), the probability that an element of a certain rank in A is NOT relevant decreases as $e^{-\Omega(\frac{n}{k}\cdot\varepsilon^2\cdot|A|)}$. Remark 22 in the end of Section 6 will sketch the ideas of deriving Lemma 17. For ease of exposition, we also use β to denote $\frac{k}{n}$ in Lemma 17 and Corollary 18.

▶ **Lemma 17.** *Let A consist of $m \le \frac{n}{4}$ elements sampled from S without replacement. Suppose that $m\beta \ge 8$ and that $\frac{1}{2} \ge \beta \ge 4\varepsilon$. Then, there is an absolute constant $\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$, coming from Theorem 26, with the following properties:*

1. *Let u be the r-th smallest element of A. If $r \le \lceil \beta m \rceil$, then u is small with probability at least*

$$\eta \cdot e^{-12\frac{\varepsilon^2}{\beta(1-\beta)}m}.$$

2. *Let v be the r-th largest element of A. If $r \le \lceil (1-\beta)m \rceil$, then v is large with probability at least*

$$\eta \cdot e^{-12\frac{\varepsilon^2}{\beta(1-\beta)}m}.$$

Since $1-\beta \ge 1/2$ and every element of the m elements is either among the $\lceil \beta m \rceil$ smallest ones or among the $\lceil (1-\beta)m \rceil$ largest ones, Lemma 17 directly implies Corollary 18.

▶ **Corollary 18.** *Let A consist of $m \le \frac{n}{4}$ elements sampled from S* without *replacement. Suppose that $m\beta \ge 8$ and that $\frac{1}{2} \ge \beta \ge 4\varepsilon$. Then, an arbitrary element u in A is NOT relevant with probability at least*

$$\eta \cdot e^{-24\cdot\frac{n}{k}\cdot\varepsilon^2\cdot m}$$

for an absolute constant $\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$ coming from Theorem 26.

By Lemma 16(3), with probability at least 1/2, the execution of $\tilde{T}$ reaches a leaf after looking at exactly $2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil$ elements. Together with Corollary 18 on each such leaf, we can lower bound the failure probability of $\tilde{T}$ as shown in the following lemma.

▶ **Lemma 19.** *If $k \ge 200$ and $4n\varepsilon \le k$, then the failure probability of $\tilde{T}$ on a uniformly shuffled input is at least*

$$\frac{1}{2} \cdot \eta \cdot e^{-24\varepsilon^2\frac{n}{k}(2\,\mathfrak{D}+\lceil\frac{8n}{k}\rceil)} \quad \text{for an absolute constant } \eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24} \text{ from Theorem 26.}$$

Proof. Recall that we build $\tilde{T}$ only when $\mathfrak{D} < \frac{n}{10}$. Fix a leaf w of $\tilde{T}$. Suppose that the execution of $\tilde{T}$ reaches w. Let x be the element that $\tilde{T}$ returns and let A be the set of elements that $\tilde{T}$ has looked at when the execution reaches w.

As $\tilde{T}$ is run on a uniformly shuffled input, the distribution of the set A as a random variable is the same as the distribution of a set of $|A|$ elements sampled from S without replacement. Note that since $\tilde{T}$ has only compared elements in A, these comparisons do not affect the distribution of A as a random variable. By Lemma 16.(1), x always has the same rank in A. If $|A| = 2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil$, then $|A| \le \frac{n}{4}$ and $\frac{k}{n} \cdot |A| \ge 8$. Moreover, we have $4n\varepsilon \le k$ by assumption. Therefore, if $|A| = 2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil$, Corollary 18 implies that $\tilde{T}$ fails with probability at least

$$\eta \cdot e^{-24 \cdot \frac{n}{k} \cdot \varepsilon^2 \cdot |A|} = \eta \cdot e^{-24 \cdot \frac{n}{k} \cdot \varepsilon^2 \cdot (2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil)}.$$

In summary, if $\tilde{T}$ reaches a leaf after looking at exactly $2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil$ elements, then $\tilde{T}$ fails with probability at least $\eta \cdot e^{-24 \cdot \frac{n}{k} \cdot \varepsilon^2 \cdot (2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil)}$. By Lemma 16.(3), the if-condition holds with probability at least $\frac{1}{2}$, leading to the statement. ◀

Since $\tilde{T}$ succeeds with probability at least $1 - Q$, we have $Q \geq \frac{1}{2}\,\eta \cdot e^{-24\varepsilon^2 \frac{n}{k}(2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil)}$, implying that $\mathfrak{D} = \Omega(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$. We can conclude the following main theorem.

▶ **Theorem 20.** *If $Q < \frac{1}{2}$, then the expected number of comparisons performed by any randomized algorithm that solves the FT-APX(k, ε) problem with probability at least $1 - Q$ is $\Omega\big(\min\big\{n, \frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q}\big\}\big)$.*

Proof. As discussed in the beginning of Section 6, if $k < 4n\varepsilon$, the lower bound $\Omega(\varepsilon^{-1} \log \frac{1}{Q})$ for approximate minimum selection [23] applies. Similarly, if $k \leq 200$, we can increase ε by $\frac{200}{n}$ so that $n\varepsilon > n \cdot \frac{200}{n} = 200 \geq k$, which changes ε by at most a constant factor[1], and apply the lower bound for the approximate minimum selection [23]. Therefore, it is sufficient to consider the case that $4\varepsilon \leq \frac{k}{n} \leq \frac{1}{2}$ and $k \geq 200$. Recall that T is the decision tree of any randomized algorithm that solves FT-APX(k, ε) with probability at least $1 - Q$ and $\mathfrak{D}$ is the expected number of elements that T looks at. If $\mathfrak{D} \geq \frac{n}{10}$, a lower bound $\Omega(n)$ follows. Otherwise, we build the auxiliary decision tree $\tilde{T}$.

By Lemma 16.(2), the success probability of $\tilde{T}$ is at least $1 - Q$, and by Lemma 19, the failure probability of $\tilde{T}$ is at least $\frac{1}{2} \cdot \eta \cdot e^{-24\varepsilon^2 \frac{n}{k}(2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil)}$ for a constant η, implying that

$$Q \geq \frac{1}{2} \cdot \eta \cdot e^{-24\varepsilon^2 \frac{n}{k}(2\,\mathfrak{D} + \lceil \frac{8n}{k} \rceil)},$$

or equivalently

$$\mathfrak{D} \geq \frac{1}{48} \cdot \frac{k}{n}\varepsilon^{-2} \ln \frac{\eta}{2Q} - \frac{1}{2}\left\lceil \frac{8n}{k} \right\rceil.$$

If $Q \leq \frac{\eta}{1000}$, since $\varepsilon^{-1} \geq \frac{4n}{k}$ (from $k \geq 4n\epsilon$), the first term $\frac{1}{48} \cdot \frac{k}{n}\varepsilon^{-2} \ln \frac{\eta}{2Q}$ dominates the second term $\frac{1}{2}\lceil \frac{8n}{k} \rceil$, and thus $\mathfrak{D} = \Omega(\frac{k}{n}\varepsilon^{-2} \ln \frac{1}{Q})$. ($\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$ as stated in Theorem 26.)

It remains to analyze the case that $Q > \frac{\eta}{1000}$, for which we construct an auxiliary algorithm that solves the FT-APX(k, ε) problem with probability at least $1 - \frac{\eta}{1000}$. We will use $\mathcal{A}$ and $\tilde{\mathcal{A}}$ to denote the original algorithm and the auxiliary algorithm, respectively. Recall that $\mathcal{A}$ solves the FT-APX(k, ε) problem with probability at least $1 - Q$. Select k' such that A outputs a small element with probability at most $\frac{k'}{n} - \frac{1-Q}{2}$ and a large element with probability at most $1 - \frac{k'}{n} - \frac{1-Q}{2}$. Thus, by using $\mathcal{A}$ to get sampled elements instead of sampling from the input, the FT-APX(k, ε) problem is reduced to the FT-APX$\big(k', \frac{1-Q}{2}\big)$ problem (with the restriction that we may only use sampled elements). Motivated by this, let $\tilde{\mathcal{A}}$ be a modified (fault-free) version of our algorithms (Section 3–5) for the FT-APX$(k', \frac{1-Q}{2})$ problem with success probability at least $1 - \frac{\eta}{1000}$ in which each sampling from S is implemented by calling $\mathcal{A}$ on S. The correctness of $\tilde{\mathcal{A}}$ relies on the fact that our algorithms only sample elements from S uniformly at random and the corresponding analysis only cares about the probability of getting a small / relevant / large element.

[1] The value of ε should be at least $\frac{1}{n}$; otherwise, the problem becomes the *exact* selection.

As applying our algorithm to solve the FT-APX($k', \frac{1-Q}{2}$) problem with probability $1 - \frac{\eta}{1000}$ would sample $O(\frac{k'}{n}(1-Q)^{-2}\log\frac{1000}{\eta})$ times from S, $\tilde{\mathcal{A}}$ invokes $\mathcal{A}$ at most $O(\frac{k'}{n}(1-Q)^{-2}\log\frac{1000}{\eta})$ times and thus performs *expected* $O(\mathfrak{D}\frac{k'}{n}(1-Q)^{-2}\log\frac{1000}{\eta})$ comparisons. Since all terms except $\mathfrak{D}$ are bounded from above by a constant, the above bound is can be reformulated as $O(\mathfrak{D})$. On the other hand, we have already proven that the expected number of comparison to solve the FT-APX(k, ε) problem with probability at least $1 - \frac{1000}{\eta}$ is $\Omega(\frac{k}{n}\varepsilon^{-2}\log\frac{1000}{\eta}) = \Omega(\frac{k}{n}\varepsilon^{-2})$. Since the first bound $O(\mathfrak{D})$ is an upper bound for the second bound $\Omega(\frac{k}{n}\varepsilon^{-2})$, $\mathfrak{D} = \Omega(\frac{k}{n}\varepsilon^{-2}) = \Omega(\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q})$. Recall that $\log\frac{1}{Q}$ is a constant since $Q \geq \frac{\eta}{1000}$ and η is an absolute constant.

To sum up, when $4\varepsilon \leq \frac{k}{n} \leq \frac{1}{2}$ and $k \geq 200$,the expected number of comparisons required by any algorithm that solve FT-APX(k, ε) with probability $1 - Q$ is

$$\Omega\left(\min\{n, \frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q}\}\right). \qquad \blacktriangleleft$$

If $\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q} = w(n)$, the lower bound in Theorem 20 becomes just $\Omega(n)$. By reducing the approximate selection problem to the exact selection problem, we can show a stronger lower bound in this case as the following theorem.

▶ **Theorem 21.** *If $Q < \frac{1}{2}$ and $\frac{k}{n}\varepsilon^{-2}\log\frac{1}{Q} = w(n)$, then the expected number of comparisons performed by any randomized algorithm that solves FT-APX(k, ε) with probability at least $1 - Q$ is*

$$\Omega\left(\max\left\{n, \varepsilon^{-1}\log\frac{\frac{k+n\varepsilon}{2n\varepsilon}}{Q}\right\}\right).$$

Proof. The first term n directly comes from the first term n of Theorem 20. Recall that we assume $k \leq \frac{n}{2}$. The second term $\varepsilon^{-1}\log\frac{(k+n\varepsilon)/(2n\varepsilon)}{Q}$ can be reduced from the lower bound $\Omega(n\log\frac{k}{Q})$ for the exact k-th smallest element selection problem [9] as follows. Note that as remarked in [9, Section 1], their bound holds both in expectation and in the worst case.

Assume we attempt to select the ℓ-th smallest element among m elements. We can duplicate each element $2 \cdot n\varepsilon$ times and solve the FT-APX$\big(k, \varepsilon\big)$ problem where $n = m \cdot n\varepsilon$ and $k = (2n\varepsilon)\cdot\ell - n\varepsilon$. This setting implies that $m = \varepsilon^{-1}$ and $\ell = (k + n\varepsilon)/(2n\varepsilon)$. Since selecting the ℓ-th smallest element among m elements with probability at least $1 - Q$ requires $\Omega(m\log\frac{\ell}{Q})$ comparisons, a lower bound of $\Omega(\varepsilon^{-1}\log\frac{(k+n\varepsilon)/(2n\varepsilon)}{Q})$ follows. $\blacktriangleleft$

▶ Remark 22. For the proof of Lemma 17, the main observation is that the number of small (or large) elements in A has a hypergeometric distribution. The probability density function of the hypergeometric distribution can be expressed explicitly with binomial coefficients. By the entropy bound for binomial coefficients and a second order tangent bound based on the second derivative, a useful tool (Theorem 32 in Appendix E.4 of the full version [17]) follows, and induces a first-tail bound for the hypergeometric distribution (Theorem 26 in Appendix D.2), from which Lemma 17 follows.

References

1 Martin Aigner. Finding the maximum and minimum. *Discrete Applied Mathematics*, 74(1):1–12, 1997.

2 Amitava Bagchi. On sorting in the presence of erroneous information. *Information Processing Letters*, 43(4):213–215, 1992.

3 Ryan S. Borgstrom and S. Rao Kosaraju. Comparison-based search in the presence of errors. In *Proceedings of the Twenty-fifth Symposium on Theory of Computing (STOC93)*, pages 130–136, 1993.

4 Mark Braverman, Jieming Mao, and S. Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the Forty-eighth48th Symposium on Theory of Computing (STOC16)*, pages 851–862, 2016.

5 Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the Nineteenth Symposium on Discrete Algorithms (SODA08)*, pages 268–276, 2008.

6 Xi Chen, Sivakanth Gopi, Jieming Mao, and Jon Schneider. Competitive analysis of the top-k ranking problem. In *Proceedings of the Twenty-Eighth Symposium on Discrete Algorithms (SODA17)*, pages 1245–1264, 2017.

7 Hyungmin Cho, Larkhoon Leem, and Subhasish Mitra. ERSA: error resilient system architecture for probabilistic applications. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 31(4):546–558, 2012.

8 Ferdinando Cicalese. *Fault-Tolerant Search Algorithms - Reliable Computation with Unreliable Information*. Monographs in Theoretical Computer Science. Springer, 2013.

9 Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

10 Irene Finocchi, Fabrizio Grandoni, and Giuseppe F. Italiano. Optimal resilient sorting and searching in the presence of memory faults. *Theoretical Computer Science*, 410(44):4457–4470, 2009.

11 Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Sorting with recurrent comparison errors. In *Proceedings of the Twenty-Eighth International Symposium on Algorithms and Computation (ISAAC17)*, pages 38:1–38:12, 2017.

12 Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Optimal sorting with persistent comparison errors. In *Proceedings of the Twenty-seventh European Symposium on Algorithms (ESA19)*, pages 49:1–49:14, 2019.

13 Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Optimal dislocation with persistent errors in subquadratic time. *Theory Comput. Syst.*, 64(3):508–521, 2020.

14 Barbara Geissmann, Matús Mihalák, and Peter Widmayer. Recurring comparison faults: Sorting and finding the minimum. In *Proceedings of the Twentieth International Symposium on Fundamentals of Computation Theory (FCT15)*, pages 227–239, 2015.

15 Ofer Grossman and Dana Moshkovitz. Amplification and derandomization without slowdown. *SIAM Journal on Computing*, 49(5):959–998, 2020.

16 Jie Han and Michael Orshansky. Approximate computing: An emerging paradigm for energy-efficient design. In *18th IEEE European Test Symposium (ETS)*, pages 1–6, 2013.

17 Shengyu Huang, Chih-Hung Liu, and Daniel Rutschman. Approximate selection with unreliable comparisons in optimal expected time. *CoRR*, abs/2205.01448, 2022. `doi:10.48550/arXiv.2205.01448`.

18 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the Thirty-nineth Symposium on Theory of Computing (STOC07)*, pages 95–103, 2007.

19 Christoph M. Kirsch and Hannes Payer. Incorrect systems: it's not the problem, it's the solution. In *Proceedings of the 49th Design Automation Conference 2012 (DAC)*, pages 913–917, 2012.

20 Rolf Klein, Rainer Penninger, Christian Sohler, and David P. Woodruff. Tolerant algorithms. In *Proceedings of the Nineteenth European Symposium on Algorithms (ESA11)*, pages 736—-747, 2011.

21 K. B. Lakshmanan, Bala Ravikumar, and K. Ganesan. Coping with erroneous information while sorting. *IEEE Transactions on Computers*, 40(9):1081–1084, 1991.

22 Tom Leighton and Yuan Ma. Tight bounds on the size of fault-tolerant merging and sorting networks with destructive faults. *SIAM Journal on Computing*, 29(1):258–273, 1999.

23 Stefano Leucci and Chih-Hung Liu. Approximate minimum selection with unreliable comparisons in optimal expected time. *Algorithmica*, 84(1):60–84, 2022.

24 Stefano Leucci, Chih-Hung Liu, and Simon Meierhans. Resilient dictionaries for randomly unreliable memory. In *Proceedings of the 27th Annual European Symposium on Algorithms, (ESA19)*, pages 70:1–70:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

25 Philip M. Long. Sorting and searching with a faulty comparison oracle. Technical report, University of California at Santa Cruz, 1992.

26 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Sorting noisy data with partial information. In *Proceedings of the Fourth Conference on Innovations in Theoretical Computer Science (ITCS13)*, pages 515–528, 2013.

27 M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2 edition, 2017.

28 Krishna Palem and Avinash Lingamneni. Ten years of building broken chips: The physics and engineering of inexact computing. *ACM Transactions on Embedded Computing Systems*, 12(2s):87:1–87:23, 2013.

29 Andrzej Pelc. Searching with known error probability. *Theoretical Computer Science*, 63(2):185–202, 1989.

30 Andrzej Pelc. Searching games with errors - fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002.

31 Bala Ravikumar, K. Ganesan, and K. B. Lakshmanan. On selecting the largest element in spite of erroneous information. In *Proceedings of the fourth Symposium on Theoretical Aspects of Computer Science (STACs87)*, pages 88–99, 1987.

32 Joseph Sloan, John Sartori, and Rakesh Kumar. On software design for stochastic processors. In *Proceedings of the 49th Annual Design Automation Conference 2012 (DAC)*, pages 918–923, 2012.

A Supplementary material for Section 2

▶ **Lemma 4** (Majority Vote). *For any error probability $p \in [0, \frac{1}{2})$, there exists a postive integer c_p such that a strategy that compares two elements $2c_p \cdot t + 1$ times and returns the majority result succeeds with probability at least $1 - e^{-t}$, where $c_p = \lceil \frac{4(1-p)}{(1-2p)^2} \rceil$. The exact failure probability of this strategy is*

$$\sum_{i=0}^{c_p \cdot t} \binom{2c_p \cdot t + 1}{i} (1-p)^i p^{2c_p \cdot t + 1 - i}.$$

Proof. Let $\{X_i \mid 1 \le i \le 2c_p \cdot t + 1\}$ be $2c_p \cdot t + 1$ *independent* Bernoulli random variables such that $X_i = 1$ if the i-th comparison succeeds, i.e., $\Pr[X_i = 1] = 1 - p$ and $\Pr[X_i = 0] = p$. Let $X = \sum_{i=1}^{2c_p \cdot t + 1} X_i$. Then, $E[X] = (2c_p \cdot t + 1)(1-p)$. Since $p < \frac{1}{2}$, we know $2(1-p) > 1$ and we can apply Lemma 5 to prove the first statement as follows:

$$\begin{aligned}
\Pr[X \le \frac{2c_p \cdot t + 1}{2}] &= \Pr[X \le \frac{1}{2(1-p)} E[X]] = \Pr[X \le \left(1 - \frac{1-2p}{2-2p}\right) E[X]] \\
&\underbrace{\le \exp\left(-\frac{1}{2} \cdot (\frac{1-2p}{2-2p})^2 \cdot E[X]\right)}_{\text{Lemma 5}} \\
&= \exp\left(-\frac{1}{2} \cdot (\frac{1-2p}{2-2p})^2 \cdot (2c_p \cdot t + 1)(1-p)\right) \\
&= \exp\left((2c_p \cdot t + 1)\frac{(1-2p)^2}{8(1-p)}\right) < \exp\left(-c_p t \frac{(1-2p)^2}{4(1-p)}\right).
\end{aligned}$$

which satisfies the statement if we choose $c_p = \lceil \frac{4(1-p)}{(1-2p)^2} \rceil$. Since X is a binomial random variable and c_p is an integer, the second statement comes as follows:

$$\Pr[X \leq \frac{2c_p \cdot t + 1}{2}] = \Pr[X \leq c_p \cdot t] = \sum_{i=0}^{c_p \cdot t} \binom{2c_p \cdot t + 1}{i} (1-p)^i p^{2c_p \cdot t + 1 - i}. \quad \blacktriangleleft$$

▶ **Lemma 5** (Chernoff Bound). *Let X be the sum of independent Bernoulli random variables. If $A \leq E[X] \leq B$, then for any $\delta \in (0,1)$,*

$$\Pr[X \geq (1+\delta) \cdot B] \leq e^{-\frac{\delta^2}{3} B} \qquad \text{and} \qquad \Pr[X \leq (1-\delta) \cdot A] \leq e^{-\frac{\delta^2}{2} A}.$$

Proof. The two statements can be extended from the proofs of [27, Theorem 4.4(2)] and [27, Theorem 4.5(2)], respectively. Here, we only state the difference. Since X is the sum of *independent* Bernoulli random variables, by [27, Section 4.2.1]

$$E[e^{tX}] \leq e^{(e^t - 1)E[X]}.$$

For the first claim, using any $\boldsymbol{t > 0}$,

$$\Pr[X \geq (1+\delta) \cdot B] = \Pr[e^{tX} \geq e^{t(1+\delta) \cdot B}] \leq \frac{E[e^{tX}]}{e^{t(1+\delta)B}} \leq \frac{e^{(e^t-1)E[X]}}{e^{t(1+\delta)B}} \overset{E[X] \leq B}{\leq} \frac{e^{(e^t-1)B}}{e^{t(1+\delta)B}}.$$

The remaining steps are identical to the proof of [27, Theorem 4.4(2)].

For the second claim, using any $\boldsymbol{t < 0}$,

$$\Pr[X \leq (1-\delta) \cdot A] = \Pr[e^{tX} \geq e^{t(1-\delta) \cdot A}] \leq \frac{E[e^{tX}]}{e^{t(1-\delta)A}} \leq \frac{e^{(e^t-1)E[X]}}{e^{t(1-\delta)A}} \overset{A \leq E[X]}{\leq} \frac{e^{(e^t-1)A}}{e^{t(1+\delta)A}}.$$

The remaining steps are identical to the proof of [27, Theorem 4.5(2)]. ◀

B Supplementary material for Section 3

▶ **Lemma 23.** *Let $m = 2^{10} \cdot 3^2 \cdot \ln \frac{2}{Q}$, let $X_1, X_2 \ldots, X_m$ be m identically and independently distributed Bernoulli random variables with probability $p \geq \frac{8}{9}$, and let $X = \sum_{i=1}^{m} X_i$.*

$$\Pr[X \geq \frac{7}{8}m] \geq 1 - \frac{Q}{2}.$$

Proof. It is sufficient to prove that $\Pr[X \leq \frac{7}{8}m] \leq \frac{Q}{2}$. Since $p \geq \frac{8}{9}$, $E[X] \geq \frac{8}{9}m$. By Lemma 5,

$$\begin{aligned} \Pr[X \leq \frac{7}{8}m] &= \Pr[X \leq (1 - \frac{1}{64}) \cdot \frac{8}{9}m] \overset{\text{Lemma 5}}{\leq} \exp\left(-\frac{1}{2} \cdot (\frac{1}{64})^2 \cdot \frac{8}{9}m\right) \\ &= \exp\left(-\frac{1}{2^{10} \cdot 3^2}m\right) \leq \exp\left(-\frac{2^{10} \cdot 3^2 \cdot \ln \frac{2}{Q}}{2^{10} \cdot 3^2}\right) = e^{-\ln \frac{2}{Q}} = \frac{Q}{2}. \end{aligned} \quad \blacktriangleleft$$

▶ **Theorem 6.** *It takes **expected** $O(\frac{k}{n}\varepsilon^{-2} \log \frac{1}{Q})$ comparisons to solve the FT-APX(k, ε) problem with probability at least $1 - Q$.*

Proof. Let $m = 2^{10} \cdot 3^2 \cdot \ln \frac{2}{Q}$ as in Lemma 23. The algorithm consists of two stages. The first stage aims to select m elements in which all elements in the range $(\frac{1}{8}m, \frac{7}{8}m]$ are relevant, and the second stage aims to select an element from $(\frac{1}{8}m, \frac{7}{8}m]$.

For the number of comparisons, by Theorem 15, it takes $O(\frac{k}{n}\varepsilon^{-2})$ comparisons to select a relevant element with probability at least $1 - \frac{1}{9}$, so the first stage takes $O(\frac{k}{n}\varepsilon^{-2} \cdot m) = O(\frac{k}{n}\varepsilon^{-2} \cdot \log \frac{1}{Q})$ comparisons. For the second stage, by Section 3, one verification step performs $O(\log \frac{1}{Q})$ comparisons. To derive the expected total number of comparisons, we need to calculate the probability of conducting the i-th round. Since the probability of picking an element in $(\frac{2}{8}m, \frac{6}{8}m]$ is $\frac{1}{2}$ at any round and such an element is verified in $(\frac{1}{8}m, \frac{7}{8}m]$ with probability at least $1 - \frac{Q}{2} \geq \frac{1}{2}$ at any round, any round returns an element with probability at least $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Similar to geometric distribution, the probability that the i-th round is conducted is at most $(1 - \frac{1}{4})^{i-1} = (\frac{3}{4})^{i-1}$, so the second stage takes expected $\sum_{i=1}^{\infty}(\frac{3}{4})^{i-1}O(\log \frac{1}{Q}) = O\big(\log \frac{1}{Q} \cdot \sum_{i=1}^{\infty}(\frac{3}{4})^{i-1}\big) = O(\log \frac{1}{Q})$ comparisons. To sum up, the algorithm takes expected $O(\frac{k}{n}\varepsilon^{-2} \cdot \log \frac{1}{Q})$ comparisons.

For the success probability, by Theorem 15 and Lemma 23, the first stage fails with probability at most $\frac{Q}{2}$. The second stage fails only when returning an element in $[1, \frac{1}{8}m]$ or $(\frac{7}{8}m, m]$. Since a single round picks an element in $[1, \frac{1}{8}m] \cup (\frac{7}{8}m, m]$ with probability $\frac{1}{4}$ and the verification fails with probability at most $\frac{Q}{2}$. a single round returns an element in $[1, \frac{1}{8}m] \cup (\frac{7}{8}m, m]$ with probability *at most* $\frac{1}{4} \cdot \frac{Q}{2} = \frac{Q}{8}$, Therefore, the failure probability of the second stage is at most $\sum_{i \geq 1}(\frac{3}{4})^{i-1} \cdot \frac{Q}{8} = \frac{Q}{2}$, concluding the statement. ◀

C Supplementary material for Section 4

▶ **Lemma 9.** *For **three** elements, consider the following median selection algorithm:*

1. *For each pair of elements, apply the majority vote strategy with $2c_p \cdot 4 + 1$ comparisons (Lemma 4), and assign a point to the element that attains the majority result.*
2. *Return the element with exactly one point. If all three elements get exactly one point, return one of them uniformly at random.*

The above algorithm returns the median with probability at least $1 - \frac{1}{13}$, and returns the minimum and the maximum with the same probability, i.e., at most $\frac{1}{26}$.

Proof. Let q be the failure probability of one majority vote. Since one majority vote consists of $2c_p \cdot 4 + 1$ comparisons, by Lemma 4, $q \leq e^{-4}$. If all three majority votes succeed, then the algorithm will return the median, implying that the algorithm will return the median with probability at least $(1-q)^3 \geq 1 - 3q \geq 1 - 3 \cdot e^{-4} \geq 1 - \frac{1}{13}$.

Now, we will prove that the algorithm returns the minimum and the maximum with the same probability. Since there are three majority votes, there are 8 possibilities, and these 8 possibilities lead to four different situations: exactly the minimum or exactly the median or exactly the maximum gets one point, or all the three elements get one point. A tree diagram for these 8 possibilities can easily calculate the probabilities of the four situations. In detail, exactly the minimum (resp. exactly the maximum) gets one point with probability $q(1-q)$, exactly the median gets one point with probability $(1-q)^3 + q^3$, and all three elements get one point with probability $q(1-q)$. Since the algorithm returns an element uniformly at random when all the three elements get one point, the algorithm returns the minimum and the maximum with the same probability $\frac{4}{3}q(1-q)$.

Since the algorithm returns the median with probability at least $1 - \frac{1}{13}$ and returns the minimum and the maximum with the same probability, the probability that the algorithm returns the minimum (resp. the maximum) is at most $\frac{1}{26}$. ◀

D Supplementary material for Section 6

D.1 Sampling Lemma

This subsection aims to build up a sampling bound (Corollary 18) that is the key ingredient to prove Lemma 19. Corollary 18 roughly states that for a set A of randomly sampled elements (without replacement), the probability that an element of a certain rank in A is NOT relevant decreases as $e^{-\Omega(\frac{\varepsilon^2}{\beta}|A|)}$. To prove Corollary 18, we first derive Lemma 17 that deals with different positions in A. For ease of exposition, we also use β to denote $\frac{k}{n}$ in the proofs. As assumed in the whole paper, $\beta \leq \frac{1}{2}$, and as stated in Section 6, it is also sufficient to consider $\beta \geq 4\varepsilon$ since if $\beta < 4\varepsilon$, we then can apply the lower bound for the approximate minimum selection [23].

▶ **Lemma 17.** *Let A consist of $m \leq \frac{n}{4}$ elements sampled from S without replacement. Suppose that $m\beta \geq 8$ and that $\frac{1}{2} \geq \beta \geq 4\varepsilon$. Then, there is an absolute constant $\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$, coming from Theorem 26, with the following properties:*

1. *Let u be the r-th smallest element of A. If $r \leq \lceil \beta m \rceil$, then u is small with probability at least*

$$\eta \cdot e^{-12\frac{\varepsilon^2}{\beta(1-\beta)}m}.$$

2. *Let v be the r-th largest element of A. If $r \leq \lceil (1-\beta)m \rceil$, then v is large with probability at least*

$$\eta \cdot e^{-12\frac{\varepsilon^2}{\beta(1-\beta)}m}.$$

Proof. We first prove (1). Let X denote the number of small elements in A. Then $X \sim \text{Hypergeom}(n, (\beta-\varepsilon)k, m)$ has a hypergeometric distribution (Definition 24 in Appendix D.2). Since $r \leq \lceil \beta m \rceil$, u is small if and only if A contains at least r small elements, i.e., if and only if $X \geq r$. Put $a = \beta$ and b $= \beta - \varepsilon$. Then we have $a \leq \frac{8}{5}b$ and $(1-a) \leq \frac{8}{5}(1-b)$ as $\beta \geq 4\varepsilon$. As $m\beta \geq 8$ and $\beta \leq \frac{1}{2}$, we also have $ma(1-a) \geq 4$. Hence by Theorem 26

$$\Pr[X \geq r] \geq \Pr[X \geq \lceil \beta m \rceil] = \Pr[X \geq \beta m] \geq \eta \cdot e^{-6\frac{\varepsilon^2}{b(1-b)}}$$

for some absolute constant $\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$. Since $\beta \geq 2\varepsilon$, we have $b \geq \frac{\beta}{2}$, and since we also have $(1-b) \geq (1-\beta)$, we have

$$\frac{\varepsilon^2}{b(1-b)} \leq \frac{2\varepsilon^2}{\beta(1-\beta)},$$

implying that

$$\Pr[X \geq r] \geq \eta \cdot e^{-12\frac{\varepsilon^2}{\beta(1-\beta)}m}$$

The proof of (2) is symmetric with large elements instead of small ones and with $(1-\beta)$ instead of β. ◀

As every element is either among the $\lceil \beta m \rceil$ smallest or among the $\lceil (1-\beta)m \rceil$ largest ones, the lemma directly implies the following.

▶ **Corollary 18.** *Let A consist of $m \leq \frac{n}{4}$ elements sampled from S* without *replacement. Suppose that $m\beta \geq 8$ and that $\frac{1}{2} \geq \beta \geq 4\varepsilon$. Then, an arbitrary element u in A is NOT relevant with probability at least*

$$\eta \cdot e^{-24 \cdot \frac{n}{k} \cdot \varepsilon^2 \cdot m}$$

for an absolute constant $\eta = \sqrt{\frac{\pi}{320}} \cdot e^{-24}$ coming from Theorem 26.

Proof. Let r be the rank of u in A. If $r \leq \lceil \beta m \rceil$, then by part (1) of Lemma 17, u is small with probability at least

$$\eta \cdot e^{-12 \frac{\varepsilon^2}{\beta(1-\beta)} m}.$$

Otherwise, $r \geq \lceil \beta m \rceil + 1 \geq \beta m + 1$, so $m + 1 - r \leq (1-\beta)m \leq \lceil (1-\beta)m \rceil$. Since u is the $(m+1-r)$-th largest element of A, by part (2) of Lemma 17, u is large with probability at least

$$\eta \cdot e^{-12 \frac{\varepsilon^2}{\beta(1-\beta)} m}.$$

Since $1 - \beta \geq \frac{1}{2}$, we have

$$\eta \cdot e^{-12 \frac{\varepsilon^2}{\beta(1-\beta)} m} \geq \eta \cdot e^{-24 \frac{\varepsilon^2}{\beta} m} = \eta \cdot e^{-24 \cdot \frac{k}{n} \cdot \varepsilon^2 \cdot m}.$$ ◀

D.2 A lower tail for hypergeometric distribution

▶ **Definition 24.** *Consider M balls, out of which K balls are black and $M - K$ balls are white.* $\mathrm{Hypergeom}(M, K, m)$ *is the probability distribution for the number of black balls in m draws from the M balls using sampling without replacement, which is the so-called* hypergeometric distribution. *$X \sim \mathrm{Hypergeom}(M, K, m)$ means that X is a random variable with $\mathrm{Hypergeom}(M, K, m)$ distribution.*

Due to the page limit, we omit the proof of Corollary 25; please see the full version [17].

▶ **Corollary 25.** *Let $X \sim \mathrm{Hypergeom}(M, K, m)$. Let $0 < \ell < m$ be an integer. Put $a = \frac{\ell}{m}$, $b = \frac{K}{M}$ and $x = \frac{m}{M}$. If $a \leq 2b$, $(1-a) \leq 2(1-b)$ and $x \leq \frac{1}{4}$, then we have*

$$Pr[X = \ell] \geq \sqrt{\frac{\pi}{64ma(1-a)}} \cdot e^{-3\frac{(a-b)^2}{b(1-b)}m}.$$

▶ **Theorem 26.** *Let $X \sim \mathrm{Hypergeom}(M, K, m)$. Let $0 \leq \ell \leq m$ be a real number with $\ell < K$ and $m - \ell < M - K$. Put $a = \frac{\ell}{m}$, $b = \frac{K}{M}$ and $x = \frac{m}{M}$. If $a \leq \frac{8}{5}b$, $(1-a) \leq 2(1-b)$, $x \leq \frac{1}{4}$ and $ma(1-a) \geq 4$, then we have*

$$Pr[X \geq \ell] \geq \sqrt{\frac{\pi}{320}} \cdot e^{-24} \cdot e^{-\frac{6(a-b)^2}{b(1-b)}m}.$$

Proof. Let $0 \leq t \leq \sqrt{ma(1-a)}$ be a real number such that $\ell + t$ is an integer and put $a' = \frac{\ell+t}{m}$. As $ma(1-a) \geq 4$, we have $t \leq \sqrt{ma(1-a)} \leq \frac{ma(1-a)}{4}$, so that

$$a' = a + \frac{t}{m} \leq a + \frac{a(1-a)}{4} \leq \frac{5}{4}a \leq 2b,$$

and $1 \leq a' \leq 1-a \leq 2(1-b)$. We may hence apply Corollary 25 and get

$$\begin{aligned}\Pr[X = \ell + t] &\geq \sqrt{\frac{\pi}{64\big(ma'(1-a')\big)}} \cdot e^{-3\frac{(a+\frac{t}{m}-b)^2}{b(1-b)}m} \\ &\geq \sqrt{\frac{\pi}{80ma(1-a)}} \cdot e^{\frac{-3(a+\frac{t}{m}-b)^2}{b(1-b)}m}\end{aligned}$$

where we used that

$$a'(1-a') \leq \frac{5}{4}a(1-a).$$

Since $(a+\frac{t}{m}-b)^2 \leq 2(a-b)^2 + 2(\frac{t}{m})^2$, we have

$$\frac{3(a+\frac{t}{m}-b)^2}{b(1-b)}m \leq \frac{6(a-b)^2}{b(1-b)}m + \frac{6t^2}{mb(1-b)}$$

where

$$\frac{6t^2}{mb(1-b)} \leq \frac{6ma(1-a)}{mb(1-b)} = 6\frac{a(1-a)}{b(1-b)} \leq 24.$$

Hence we have

$$\Pr[X = \ell + t] \geq \sqrt{\frac{\pi}{80\big(ma(1-a)-t\big)}} \cdot e^{-\frac{6(a-b)^2}{b(1-b)}m} \cdot e^{-24}.$$

There are at least $\sqrt{ma(1-a)} - 1$ possible values of t. As $ma(1-a) \geq 4$, we have

$$\sqrt{ma(1-a)} - 1 \geq \frac{\sqrt{ma(1-a)}}{2}.$$

Thus summing over all possible possible values of t yields the statement. ◀