A Distributional Analysis of QuickXsort for Mergesort

Jasper Ischebeck
Department of Mathematics, Uppsala University, Sweden

Florian Lesny
Institute of Mathematics, Goethe University Frankfurt, Frankfurt a.M., Germany

Ralph Neininger[1]
Institute of Mathematics, Goethe University Frankfurt, Frankfurt a.M., Germany

Abstract

QuickXsort is an efficient in situ sequential sorting algorithm that mixes Hoare's Quicksort algorithm with another sorting algorithm X, such as Heapsort, Insertionsort or Mergesort. The advantage is that QuickXsort can be in-place even if X is not. QuickXsort works recursively like Quicksort but uses sorting algorithm X on one of the sub-lists generated in each step. While the expected complexity of QuickXsort, measured by the number of key comparisons, has been investigated for various choices of X, here the asymptotic variance and distribution of the normalized complexity are studied with Mergesort used as X. Various versions of Mergesort and splitting regimes for the decomposition of the list by Quicksort are considered and periodicities in moments and the distributions are characterized.

2012 ACM Subject Classification Theory of computation → Sorting and searching

Keywords and phrases QuickXsort, Analysis of Algorithms, Mergesort, distribution analysis, limit law, weak convergence, contraction method

Digital Object Identifier 10.4230/LIPIcs.AofA.2026.22

Funding *Jasper Ischebeck*: Ragnar Söderberg Foundation and Göran Gustafsson Foundation

Acknowledgements We thank Sebastian Wild for pointing out the present problem and for addressing the subject during his plenary talk at the Analysis of Algorithms meeting 2025 in Toronto.

1 Introduction

Sorting is heavily used in practice; therefore, it is important to use fast sorting methods to save time. One problem that arises in many cases is that these fast algorithms often do not work in-place, so external space is needed, which decreases the usability of these methods. One way of solving this problem is the use of QuickXsort, initially investigated by [3] in the sample case of QuickHeapsort. QuickXsort is a mixed sorting algorithm for an array that uses the ordinary Quicksort method for partitioning the array, but then one of the segments (also called sublists) is sorted by another sorting algorithm, X. This gives us the ability to obtain internal sorting methods that are still comparison-optimal up to lower terms.

QuickXsort. For sorting algorithm X, we consider algorithms that are not in-place, but need some linear buffer space. More specifically, for $\alpha \in [0, 1]$, they require additional space of size $\lfloor \alpha n \rfloor$ to sort a list of n elements. Therefore, a larger α value corresponds to algorithm X requiring more buffer space. Furthermore, we assume that after X has sorted the elements in its segment, the buffer contains the same elements as before, possibly in a different order.

[1] Corresponding author

37th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2026).
Editor: Konstantinos Panagiotou; Article No. 22; pp. 22:1–22:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

QuickXsort then works in the following way: At first, we choose a pivot element by selecting $(2h-1)$ elements uniformly at random, where $h \in \mathbb{N} = \{1, 2, \ldots\}$ is fixed. The median of these elements is used as the pivot. We then divide our list as in the classical Quicksort algorithm, by comparing each element with the pivot to obtain two segments. The first segment contains all elements smaller than the pivot, and the second segment contains all elements greater than the pivot. If possible, we now sort the larger of the two segments with algorithm X, using the smaller segment as a buffer. This is possible if and only if the size of the smaller segment, I_n, fulfills the condition $I_n \geq \alpha(n - I_n - 1)$, since algorithm X requires buffer space equal to $\lfloor \alpha(n - I_n - 1) \rfloor$ to sort a list of $n - I_n - 1$ elements. If this condition is not satisfied, we use X on the smaller segment instead, which is always possible since $\alpha \leq 1$. The other segment, which is not sorted by X, is then recursively sorted using the QuickXsort algorithm. So, after one recursion step, the X-sorted segment is in the right order, while the elements of the unsorted segment are in arbitrary order.

The advantage of this new sorting method is that X can use the second segment as a buffer, as long as the elements originally stored in the second segment are still intact after the sorting procedure X. This leads to more efficient sorting algorithms, which still sort in-place. Suitable sorting methods for X are Mergesort and Heapsort. For QuickMergesort, for which $\alpha = 1/2$, the condition $I_n \geq (n - I_n - 1)/2$ is implied. Therefore, Mergesort is used on the larger segment if it contains no more than 2/3 of all elements. For more information on QuickXSort, see also [8].

In [8], the expected number of key comparisons required is analyzed for the random model of sorting a uniformly distributed permutation, which will be assumed throughout; explicitly, the leading constant in the expectation of QuickXsort is derived from the leading constant of the sorting algorithm X. If X is Mergesort, we thus have the optimal $n \log_2 n$. We refer to and comment on the results in [8] below.

Mergesort, attributed to von Neumann, cf. [19] and [20], is a sorting algorithm where the list is split into two parts; these two parts are sorted recursively, and then the two sorted sublists are merged into a single, sorted list. Versions of Mergesort differ in the way the list is split and in how the lists are merged. Its advantage compared to Quicksort is its comparison-optimality in the leading term; however, in return it requires a buffer.

In the present paper, we consider two ways to split the lists: Top-down Mergesort and Queue-Mergesort. In Top-down Mergesort, the list is split up evenly, with one list of size $\lceil \frac{n}{2} \rceil$ and one of size $\lfloor \frac{n}{2} \rfloor$. In Queue-Mergesort, the procedure starts from the bottom, with every element forming a single list of size 1, and then the two smallest lists are merged iteratively. This may lead to an uneven split in the top, where the second list can be up to two times smaller than the larger list. In fact, the size of the larger sublist is the unique power of 2 in $\left(\frac{1}{3}n, \frac{2}{3}n\right]$. Details are given in Section 2.

After sorting the sublists, they are merged. This merging step cannot be done efficiently in-place for arrays, and a buffer is needed. The usual implementations require a buffer as large as the smaller list, so at most $\lfloor \frac{n}{2} \rfloor$. Therefore, we have $\alpha = \frac{1}{2}$ (recall that the size of the buffer required is denoted by $\lfloor \alpha n \rfloor$), and only segments with at most $\frac{2}{3}$ of the elements can be sorted using Mergesort. The alternative merge algorithm in [25] requires only half the size of the smaller list as a buffer, so $\alpha = \frac{1}{4}$, and α can be further improved by more sophisticated merge operations. However, for all constants explicitly calculated in this paper, we will assume that $\alpha = \frac{1}{2}$.

Size of segments. Let P_h be a beta(h,h)-distributed random variable. If the pivot is chosen as the median of a random subsample of size $2h-1$, then the size J_n of the left sublist (containing the elements smaller than the pivot) fulfills $J_n/n \to P_h$ in distribution and with all moments. Let I_n denote the size of the sublist recursively sorted by the QuickXsort algorithm. Then, by the continuous mapping theorem, we get

$$\frac{I_n}{n} \to Z, \tag{1}$$

in distribution and with all moments, where Z is a random variable with a distribution denoted by $\varrho = \mathcal{L}(Z)$ having density

$$f_\varrho(x) := \frac{2}{B(h,h)} x^{h-1}(1-x)^{h-1}\left(1_{(1/3,1/2)} + 1_{(2/3,1)}\right)(x), \quad x \in [0,1]. \tag{2}$$

Here, $B(\,\cdot\,,\,\cdot\,)$ denotes the beta function (Euler integral). In the case of $h=1$, where the pivot is uniformly distributed, we have

$$f_\varrho(x) = 2 \cdot (\mathbf{1}_{(1/3,1/2)} + \mathbf{1}_{(2/3,1)})(x), \quad x \in [0,1]. \tag{3}$$

▶ Remark 1. There is another common version of Mergesort called Bottom-up Mergesort, where the size of the larger sublist is the largest power of two less than n. The number of comparisons has a variance of order $\Omega(n^2)$ as can easily be seen by considering the case of $n = 2^m + 1$ for $m \in \mathbb{N}$. A limit law after normalization for the number of comparisons of Bottom-up Mergesort is unknown. We do not consider Bottom-up Mergesort in the present extended abstract.

▶ Remark 2. For QuickXsort, the case of h depending on n of order n^β for $\beta \in (0, \frac{1}{2}]$ has also been considered. We hope to report on a distributional analysis for such cases in the full paper version of the present extended abstract.

The present extended abstract is organized as follows. In Section 2 we recall the analysis of the complexity of Mergesort as far as it is needed subsequently.

Section 3 contains our main results on the complexity of QuickXsort for Mergesort measured by the number of key comparisons. There, the second order periodic term in the expectation is given, along with the periodic first order term of the variance and the periodic behavior of the distribution after normalization, see Theorem 6. Fourier series for the appearing periodic functions are given in Theorem 7. For the case of median-of-1 and median-of three versions, bounds on the expansions of expectation and variance are given in Corollaries 8 und 9.

In Section 4 we develop a version of the contraction method for periodic toll terms, see Theorem 11, on which our analysis is based. We feel that Theorem 11 may also be useful for further analysis and other problems.

Section 5 contains proofs of our main results on QuickXsort. The results on and following from Fourier series representations of the periodic functions (Theorem 7 and Corollary 9) are only stated here and will be included in the full paper version.

Our approach seems to be robust and may also be applicable to other choices of X. We comment on this in Remark 10.

2 Mergesort

In this section, we recall results on the average case behavior of Mergesort which are required subsequently.

Top-down Mergesort. Top-down Mergesort proceeds in the following way: The initial list is divided in two sublists of size $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ and Mergesort is recursively applied on both sublists. Two sorted list are merged by comparing the smallest elements of the sublists, removing the smaller of the two to some auxiliary space and proceeding until one of the two sublists is empty. In [14], expressions for the mean and variance are derived. The function Π^{td}, defined by

$$\Pi^{td}(x) = \frac{1}{2} - \frac{1}{\log(2)} + \frac{\Psi'(0)}{\log(2)} + \frac{1}{\log(2)} \sum_{k \in \mathbb{Z}\setminus\{0\}} \frac{1 + \Psi(\chi_k)}{\chi_k\,(\chi_k + 1)} e^{2k\pi i x} \tag{4}$$

with $\chi_k = \frac{2k\pi i}{\log(2)}$ and

$$\Psi(u) = \sum_{m \geq 1} \frac{2}{(m+1)(m+2)} \left(-\frac{1}{(2m)^u} + \frac{1}{(2m+1)^u} \right) \tag{5}$$

plays a fundamental role in expressing the periodic linear term in the expectation. The function Π^{td} is continuous with a period of 1.

▶ **Lemma 3** (Hwang [14]). *Let M_n^{td} be the number of key comparisons used by Top-down Mergesort to sort a list of n uniformly permuted elements. Then,*

$$\mathbb{E}\left[M_n^{td}\right] = n\log_2(n) + n\Pi^{td}\left(\log_2(n)\right) + 2 - \sum_{k \geq 0} \frac{1}{2^k\left(2^k n + 1\right)}, \quad n \geq 1, \tag{6}$$

and $\operatorname{Var}\left(M_n^{td}\right) = \Theta(n)$ *as* $n \to \infty$.

Queue-Mergesort. Another variant of Mergesort is Queue-Mergesort, proposed and first analysed with respect to worst case behaviour by [12]. In this variant, the elements are first divided into lists of size 1 and then arranged as a queue. Then, the first two lists of the queue are merged by a linear merge algorithm (e.g. in the same way as described in the section on Top-down Mergesort). The new list is then put at the tail of the queue and the procedure is repeated until there remains only one sorted list, see [12] for details. An advantage of this variant over Top-down Mergesort is the smaller linear term in the variance, on the contrary, the linear term in the expectation of Top-down Mergesort is better than in Queue-Mergesort, see [4, Table 1]. The periodic function Π^q, which appears in the expectation of Queue-Mergesort, has the form

$$\Pi^q(x) = 1 - \{x - \beta\} - 2^{1-\{x-\beta\}}, \tag{7}$$

where $\beta = \sum_{j \geq 0} \frac{1}{2^j+1} - 1$ is a constant and $\{x\}$ for $x \in \mathbb{R}$ denotes the fractional part, i.e., $\{x\} := x - \lfloor x \rfloor$. Then we have:

▶ **Lemma 4** (Chen, Hwang and Chen [4]). *Let M_n^q be the number of key comparisons Queue-Mergesort requires to sort a uniformly permuted list of n elements. Then,*

$$\mathbb{E}\left[M_n^q\right] = n\log_2(n) + \left(\Pi^q(\log_2(n) - \beta\right)n + \mathrm{O}(\log_2 n) \tag{8}$$

with $\beta = \sum_{j \geq 0} \frac{1}{2^j+1} - 1$. *Furthermore, we have* $\operatorname{Var}(M_n^q) = \Theta(n)$ *as* $n \to \infty$.

For $x \in [0, 1)$, the function $\Pi^q(x)$ is minimal at 0 with $\Pi^q(0) = -1$ and maximal at $\log_2(\ln 2)$ with $\Pi^q(\log_2(\ln 2)) = -\frac{1+\ln\ln 2}{\ln 2}$, so Π^q maps into $[-0.9570 + 0.0430, -0.9570 - 0.0430]$. The Fourier series $x \mapsto \sum_{k\in\mathbb{Z}} c_{\Pi^q,k} e^{2\pi i k x}$ of Π^q converges absolutely and is given by the coefficients,

$$c_{\Pi^q,k} = \int_0^1 (1 - y - 2^{1-y}) e^{-2\pi i k y - 2\pi i k \beta} \mathrm{d}y = \frac{e^{-2\pi i k \beta}}{2\pi k(i - 2\pi k \ln 2)}. \tag{9}$$

▶ Remark 5. In [13, 14, 4], more refined results than those stated in Lemma 3 and Lemma 4 are given, where we state results of [14, 4] only up to a precision needed for our subsequent analysis. In fact, the expansions of the expectations in Lemmas 3 and 4 are only needed up to a $o(n)$ precision here.

3 Results

QuickMergesort was first analysed in [7] with results expanded and sharpened in [8]. Since Mergesort has a periodic second order linear term in its expectation, a sort of distributional periodicity occurs in the complexity of QuickMergesort.

For $\Pi : \mathbb{R} \to \mathbb{R}$ being a continuous function with period 1 and $(Z_j)_{j\geq 1}$ a sequence of i.i.d. random variables with a given probability distribution ϱ on $(0,1)$, we define the family $\{\Upsilon(x) \,|\, x \in [0,1]\}$ of real random variables $\Upsilon(x) = \Upsilon^{\Pi,\varrho}(x)$ by

$$\Upsilon(x) = \sum_{k=1}^{\infty} \left(\prod_{j=1}^{k-1} Z_j \right) \left(\mathcal{E}(Z_k) + (1 - Z_k)\Pi\left(x + \log_2(1 - Z_k) + \sum_{j=1}^{k-1} \log_2(Z_j) \right) \right), \tag{10}$$

with

$$\mathcal{E}(y) := y \log_2(y) + (1 - y) \log_2(1 - y) + 1, \quad y \in (0,1), \tag{11}$$

and empty products are considered to be 1, and empty sums are 0. The proof of Theorem 11 below covers the fact that the $\Upsilon(x)$ are almost surely finite random variables for all such Π, ϱ and x.

The ℓ_2 metric appearing in the following theorem is defined below in (27); it is stronger than weak convergence.

▶ **Theorem 6.** *For the number X_n of key comparisons required by QuickMergesort to sort a list of n uniformly permuted numbers with pivots chosen as medians of subsamples of size $2h - 1$ with $h \in \mathbb{N}$ and the versions Top-down Mergesort or Queue-Mergesort, we have with Υ from (10) with ϱ given by (2) and*

$\Pi = \Pi^{td}$ *given in* (4) *for Top-down Mergesort,*

$\Pi = \Pi^{q}$ *given in* (7) *for Queue-Mergesort,*

that

$$\ell_2 \left(\frac{X_n - n \log_2 n}{n}, \Upsilon(\{\log_2 n\}) \right) \to 0 \quad (n \to \infty). \tag{12}$$

In particular, for any subsequence of the form $n_k = 2^{j_k + x + \varepsilon_k}$ with $j_k \in \mathbb{N}$, $x \in [0,1]$, and $\varepsilon_k \to 0$, we have convergence in distribution

$$\frac{X_{n_k} - n_k \log_2 n_k}{n_k} \xrightarrow{d} \Upsilon(x) \quad (k \to \infty). \tag{13}$$

Moreover, we obtain, as $n \to \infty$, that

$$\mathbb{E}[X_n] = n \log_2(n) + \Lambda_E(\log_2 n)n + o(n), \qquad \mathrm{Var}(X_n) = \Lambda_V(\log_2 n)n^2 + o(n^2), \tag{14}$$

with continuous, 1-periodic functions $\Lambda_E(x) := \mathbb{E}[\Upsilon(x)], \Lambda_V(x) := \mathrm{Var}(\Upsilon(x))$ for $x \in [0,1]$.

The functions Λ_E and Λ_V are most naturally expressed as Fourier series, which are characterised in the subsequent Theorem 7. We denote by $c_{\Pi,k}$ the k-th Fourier coefficient of Π,

given by (4) for Top-down Mergesort,
given by (9) for Queue-Mergesort.

Furthermore, we denote the k-th Fourier coefficient of Λ_E by $c_{E,k}$ and the k-th Fourier coefficient of Λ_V by $c_{V,k}$. We set

$$\chi_k := \frac{2\pi i k}{\ln 2}$$

and recall $\mathcal{E}$ defined in (11).

▶ **Theorem 7.** *Under the conditions of Theorem 6, the Fourier series of Λ_E and Λ_V converge absolutely, i.e., we have*

$$\Lambda_E(x) = \sum_{k\in\mathbb{Z}} c_{E,k} e^{2k\pi i x}, \qquad \Lambda_V(x) = \sum_{k\in\mathbb{Z}} c_{V,k} e^{2k\pi i x}. \tag{15}$$

Moreover, with Z having distribution ϱ, we have

$$c_{E,k} = \mathbf{1}(k=0)\frac{\mathbb{E}[\mathcal{E}(Z)]}{1-\mathbb{E}[Z]} + \frac{\mathbb{E}\big[(1-Z)^{1+\chi_k}\big]}{1-\mathbb{E}[Z^{1+\chi_k}]} c_{\Pi,k}. \tag{16}$$

The function Λ_E is smooth (infinitely differentiable). Furthermore,

$$c_{V,k} = \frac{1}{1-\mathbb{E}[Z^{2+\chi_k}]} \sum_{l\in\mathbb{Z}} (\mathbb{E}[c_{Z,l}c_{Z,k-l}] - \mathbb{E}[c_{Z,l}]\mathbb{E}[c_{Z,k-l}]), \tag{17}$$

where $c_{Z,k}$ for $k \in \mathbb{Z}$ are the random variables

$$c_{Z,k} := \mathbf{1}(k=0)\left(\frac{\mathbb{E}[\mathcal{E}(Z)]}{1-\mathbb{E}[Z]} Z + \mathcal{E}(Z)\right) + c_{\Pi,k}\left((1-Z)^{1+\chi_k} + \frac{\mathbb{E}\big[(1-Z)^{1+\chi_k}\big]}{1-\mathbb{E}[Z^{1+\chi_k}]} Z^{1+\chi_k}\right). \tag{18}$$

We use the notation $(\kappa \pm \lambda)b_n$ to denote a term, say a_n, with $a_n \in [(\kappa-\lambda)b_n, (\kappa+\lambda)b_n]$. By bounding the function Λ_E, we obtain bounds on $\mathbb{E}[X_n]$. Recall, that we assume $\alpha = \frac{1}{2}$.

▶ **Corollary 8.** *For the X_n from Theorem 6 using Top-down Mergesort, we have*

$$\mathbb{E}[X_n] = n\log_2 n - (0.3407 \pm 0.01185)n + o(n) \quad \textit{for median-of-1 and} \tag{19}$$
$$\mathbb{E}[X_n] = n\log_2 n - (0.8477 \pm 0.01185)n + o(n) \quad \textit{for median-of-3.} \tag{20}$$

For Queue-Mergesort,

$$\mathbb{E}[X_n] = n\log_2 n - (0.0450 \pm 0.04303)n + o(n) \quad \textit{for median-of-1 and} \tag{21}$$
$$\mathbb{E}[X_n] = n\log_2 n - (0.5520 \pm 0.04303)n + o(n) \quad \textit{for median-of-3.} \tag{22}$$

Note that the bound (20) has already been shown in [8, Corollary 5.1].

We can bound the variance in the same way:

▶ **Corollary 9.** *For the X_n from Theorem 6 using Top-down Mergesort, we have*

$$\mathrm{Var}(X_n) = (0.4283 \pm 0.01551)n^2 + o(n^2) \quad \textit{for median-of-1 and} \tag{23}$$
$$\mathrm{Var}(X_n) = (0.1069 \pm 0.00775)n^2 + o(n^2) \quad \textit{for median-of-3.} \tag{24}$$

For Queue-Mergesort,

$$\mathrm{Var}(X_n) = (0.4300 \pm 0.05632)n^2 + o(n^2) \quad \textit{for median-of-1 and} \tag{25}$$
$$\mathrm{Var}(X_n) = (0.1086 \pm 0.02813)n^2 + o(n^2) \quad \textit{for median-of-3.} \tag{26}$$

▶ Remark 10. Unlike Mergesort's asymptotic normal behaviour, the limiting behaviour in Theorem 1 does not have components of a normal distribution. This is because our quadratic variance is caused by uneven splitting in the Quicksort decomposition with pivot elements. This quadratic variance masks the smaller normal fluctuations in Mergesort's contribution, meaning they are no longer visible in the distributional limit behaviour. Instead, the splitting and the expectation for Mergesort determine the limiting behaviour. For this reason, we believe that our proof technique is robust with respect to the sorting procedure X, as it only requires information on the first two moments up to the precision of Lemmas 3 and 4, cf. Remark 5.

4 A contraction method for periodic toll functions

As a tool for our proofs, we develop a contraction method for periodic toll functions. For a general introduction to the contraction method we refer to [27, 22, 23].

For $p \geq 1$ let $\mathcal{P}_p$ denote the space of all probability distributions $\mu = \mathcal{L}(X)$ with a finite p-th absolute moment, i.e., with $\|X\|_p := (\mathbb{E}|X|^p)^{1/p} < \infty$. The minimal L_p metrics ℓ_p are given by

$$\ell_p(\mu, \nu) := \inf\left\{\|X - Y\|_p \;:\; \mathcal{L}(X) = \mu,\, \mathcal{L}(Y) = \nu\right\}, \quad \mu, \nu \in \mathcal{P}_p. \tag{27}$$

It is known that $(\mathcal{P}_p, \ell_p)$ are complete metric spaces and convergence in ℓ_p is equivalent to weak convergence plus convergence of the p-th absolute moment. For $\mu, \nu \in \mathcal{P}_p$ there always exist random variables X, Y on a joint probability space with $\mathcal{L}(X) = \mu$, $\mathcal{L}(Y) = \nu$ and $\ell_p(\mu, \nu) = \|X - Y\|_p$. Such random variables are called optimal couplings of μ and ν. It is also possible to optimally couple an arbitrary subset $\mathcal{Q} \subset \mathcal{P}_p$, i.e., to find random variables X_μ on a joint probability space with $\mathcal{L}(X_\mu) = \mu$ for all $\mu \in \mathcal{Q}$ such that X_μ, X_ν are optimal couplings for all $\mu, \nu \in \mathcal{Q}$. For these and further properties of the minimal L_p metric ℓ_p see [21, 2, 24, 1, 29]. We write short $\ell_p(X, Y) := \ell_p(\mathcal{L}(X), \mathcal{L}(Y))$.

The periodic behaviour of moments and distributions is well-known in the analysis of random discrete structures and in the analysis of algorithms. E.g., periodic functions are present in the moments of random variables that appear in the context of digital search trees, tries or PATRICIA tries. In such cases, after subtracting the expectation and dividing by the standard deviation, normal limit laws often appear; see, e.g., [20, 10, 28, 23, 11, 17, 15]. There are also periodicities in the area of random discrete structures that prevent convergence in distribution after exact normalization but exhibit convergence along specific subsequences, see, e.g., [6, 9, 16, 18, 5].

The periodic behavior appearing in Theorem 11 (and hence, as an application, in our main Theorem 6) seems to be of a different nature. We hence provide a general framework to cover such cases.

Subsequently, if random variables X, Y have the same distribution, we write $X \stackrel{d}{=} Y$.

Let $(Y_n)_{n\geq 0}$ be a sequence of random variables fulfilling the recursion

$$Y_n \stackrel{d}{=} A_n Y_{I_n} + b_n, \quad n \geq n_0, \tag{28}$$

where I_n is a random integer in $\{0, \ldots, n\}$ and $n_0 \in \mathbb{N}$. Furthermore, A_n, b_n are random variables such that the vector (I_n, A_n, b_n) is independent of $(Y_0, Y_1, \ldots, Y_n)$. We assume that all random variates have finite absolute moments of order p with a $p \geq 1$, i.e., their distributions are in $\mathcal{P}_p$.

In classical theorems of the contraction method, it is assumed that (A_n, b_n) converges in ℓ_p to some limit (A, b) with $\|A\|_p < 1$. Then, under some technical conditions, convergence of Y_n in ℓ_p can be shown towards the within $\mathcal{P}_p$ unique solution of the equation

$$Y \stackrel{d}{=} AY + b, \tag{29}$$

where, on the right hand side, Y is independent of (A, b), see [26, 22].

However, in the examples of the present extended abstract, we do not have the convergence $b_n \to b$, and $Y_n \to Y$ fails to hold. Instead, to cover our QuickXSort cases, in the following theorem, we assume a periodic behavior of b_n, which is then reflected in the asymptotic behavior of Y_n.

▶ **Theorem 11.** *Let $(Y_n)_{n \geq 0}$ be a sequence of random variables fulfilling the recursion*

$$Y_n \stackrel{d}{=} A_n Y_{I_n} + b_n, \tag{30}$$

with conditions as stated after (28). Assume there exist random variables A, Z, and $\{b^\star(x) \mid x \in [0,1]\}$, all L_p-integrable, such that $x \mapsto b^\star(x)$ is continuous (for all realizations), and a real number $d > 0$ with

$$\ell_p\Big((b_n, A_n, I_n/n), (b^\star(\{\log_d n\}), A, Z)\Big) \to 0 \quad (n \to \infty), \tag{31}$$

$$\|A\|_p < 1, \tag{32}$$

$$b^\star(0) = b^\star(1) \ \textit{a.s.}, \tag{33}$$

$$\lim_{y \to x} \|b^\star(x) - b^\star(y)\|_p = 0 \textit{ for all } x \in [0,1], \tag{34}$$

$$\mathbb{E}\big[|A_n|^p \mathbf{1}_{\{I_n \leq k\} \cup \{I_n = n\}}\big] \to 0 \quad (n \to \infty) \textit{ for all } k \in \mathbb{N}. \tag{35}$$

Then, subject to $\sup_{x \in [0,1]} \|\Upsilon(x)\|_p < \infty$ and Υ having continuous paths (i.e., $x \mapsto \Upsilon(x)$ is continuous for all realizations), there exists a solution of

$$\Upsilon(x) \stackrel{d}{=} A\Upsilon(\{x + \log_d(Z)\}) + b^\star(x), \quad x \in [0,1], \tag{36}$$

whose one-dimensional marginals are uniquely determined. Furthermore, we have

$$\ell_p(Y_n, \Upsilon(\{\log_d n\})) \longrightarrow 0 \quad \textit{for } n \to \infty. \tag{37}$$

Moreover, $\Upsilon(0) \stackrel{d}{=} \Upsilon(1)$ and $\lim_{y \to x} \ell_p(\Upsilon(x), \Upsilon(y)) = 0$ for all $x \in [0,1]$.

▶ Remark 12. If b_n converges in ℓ_p, we can choose $b^\star(x)$ independently of x. Then $\Upsilon(\{\log_d n\})$ also does not depend on n, and we recover a basic ℓ_p contraction theorem.

Proof of Theorem 11. We start by proving that $\Upsilon(x)$ exists. By recursively expanding (36), we obtain the form

$$\Upsilon(x) \stackrel{d}{=} \sum_{i=0}^{\infty} b^{\star,(i)} \left(\left\{ x + \sum_{j=0}^{i-1} \log_d(Z_j) \right\} \right) \prod_{j=0}^{i-1} A^{(j)}, \tag{38}$$

where $(A^{(j)}, b^{\star,(j)}, Z_j)$ are independent copies of $(A, b^\star, Z)$. Note that $x \mapsto \|b^\star(x)\|_p$ is continuous by (34), hence it is bounded. The L_p norms of the summands in (38) are bounded by $\|A\|_p^i \sup_{x\in[0,1]} \|b^\star(x)\|_p$ due to independence. Hence, by (32) there appears a dominating geometric series which implies

$$\sup_{x\in[0,1]} \|\Upsilon(x)\|_p \le \sup_{x\in[0,1]} \frac{\|b^\star(x)\|_p}{1-\|A\|_p}. \tag{39}$$

Now, we choose $\Upsilon(x)$ as the versions given by the right hand sides of (38), i.e. pointwise and not only in distribution. From (38) we see that $\Upsilon(0) = \Upsilon(1)$ and that $\lim_{y\to x} \|\Upsilon(x), -\Upsilon(y)\|_p = 0$ for all $x \in [0,1]$ in view of (34). Since Υ is defined on a compact interval,

$$\omega(\delta) := \sup_{|x-y|\le\delta} \big\|\Upsilon(x) - \Upsilon(y)\big\|_p \tag{40}$$

is a bounded function on $[0,1]$ with $\omega(\delta) \to 0$ for $\delta \to 0$. Because Z and I_n are independent of $\Upsilon(x)$, we have

$$\mathbb{E}\Big[\Big|\Upsilon(\log_d(nZ)) - \Upsilon(\log_d(I_n))\Big|^p \;\Big|\; Z, I_n\Big] \le \omega^p\big(|\log_d(nZ) - \log_d(I_n)|\big). \tag{41}$$

Therefore, using independence of Υ from (I_n, A, Z),

$$\mathbb{E}\Big[\big|A\big(\Upsilon(\{\log_d I_n\}) - \Upsilon(\{\log_d(nZ)\})\big)\big|^p\Big] \le \mathbb{E}\big[|A|^p \omega^p\big(|\log_d(nZ) - \log_d(I_n)|\big)\big] \to 0 \tag{42}$$

because $\log_d I_n - \log_d(nZ) \to 0$ in probability and $|A|^p \sup_{\delta\in[0,1]} \omega^p(\delta)$ is a integrable dominating function.

To show uniqueness, assume $\{\mathcal{L}(\widetilde{\Upsilon}(x)), x \in [0,1]\}$ is also a solution of system (36). We fix $(A, Z, b^\star(x))$ as in (36) and may assume that the pairs $(\widetilde{\Upsilon}(x), \Upsilon(x))$ as optimal couplings of their distributions and such that the family $\{(\widetilde{\Upsilon}(x), \Upsilon(x)) \,|\, x \in [0,1]\}$ is independent of $(A, Z, b^\star(x))$. Thus, we have

$$\Upsilon(x) \stackrel{d}{=} A\Upsilon(\{x + \log_d(Z)\}) + b^\star(x), \quad x \in [0,1], \tag{43}$$

$$\widetilde{\Upsilon}(x) \stackrel{d}{=} A\widetilde{\Upsilon}(\{x + \log_d(Z)\}) + b^\star(x), \quad x \in [0,1]. \tag{44}$$

Note that

$$c := \sup_{x\in[0,1]} \ell_p(\Upsilon(x), \widetilde{\Upsilon}(x)) \le \sup_{x\in[0,1]} \|\Upsilon(x)\|_p + \sup_{x\in[0,1]} \|\widetilde{\Upsilon}(x)\|_p < \infty.$$

Using (43), (44) and that $(\widetilde{\Upsilon}(x), \Upsilon(x))$ are optimal couplings we obtain

$$\begin{aligned}
\ell_p^p\Big(\Upsilon(x), \widetilde{\Upsilon}(x)\Big) &\le \mathbb{E}\Big[\Big|A\big(\Upsilon(\{x + \log_d(Z)\}) - \widetilde{\Upsilon}(\{x + \log_d(Z)\})\big)\Big|^p\Big] \\
&\le \mathbb{E}\Big[|A|^p \mathbb{E}\Big[\Big|\Upsilon(\{x + \log_d(Z)\}) - \widetilde{\Upsilon}(\{x + \log_d(Z)\})\Big|^p \;\Big|\; Z\Big]\Big] \\
&\le \mathbb{E}[|A|^p] c^p.
\end{aligned}$$

Since $\mathbb{E}[|A|^p] < 1$, this implies $c = 0$ and thus $\Upsilon(x) \stackrel{d}{=} \widetilde{\Upsilon}(x)$ for all $x \in [0,1]$.

We subsequently use the short notation $Y_n^\star := \Upsilon(\{\log_d n\})$, $b_n^\star := b(\{\log_d n\})$. Note that we can choose (b_n, A_n, I_n) and $(b_n^\star, A, Zn)$ such that componentwise we have optimal couplings. Moreover, we choose $(Y_j, Y_j^\star)$ as optimal couplings such that these pairs are independent of $(b_n, A_n, I_n, b_n^\star, A, Zn)$. Note that we have

$$Y_n \stackrel{d}{=} A_n Y_{I_n} + b_n, \qquad Y_n^\star \stackrel{d}{=} AY_{Zn}^\star + b_n^\star.$$

Hence,

$$\begin{aligned}
\Delta(n) &:= \ell_p(Y_n, Y_n^\star) \\
&\leq \|A_n Y_{I_n} - AY^\star_{Zn} + b_n - b_n^\star\|_p \\
&\leq \big\|A_n\big(Y_{I_n} - Y^\star_{I_n}\big)\big\|_p + \big\|(A_n - A)Y^\star_{I_n}\big\|_p + \big\|A(Y^\star_{I_n} - Y^\star_{nZ})\big\|_p + \|b_n - b_n^\star\|_p \\
&\leq \mathbb{E}\big[|A_n|^p\big(Y_{I_n} - Y^\star_{I_n}\big)^p\big]^{1/p} + \|A_n - A\|_p \sup_{m\in\mathbb{N}} \|Y^\star_m\|_p + \big\|A(Y^\star_{I_n} - Y^\star_{nZ})\big\|_p + \|b_n - b_n^\star\|_p \\
&= \mathbb{E}[|A_n|^p \Delta^p(I_n)]^{1/p} + o(1), \qquad (45)
\end{aligned}$$

where the last three summands are $o(1)$ because of (31), the perfect coupling and (42).

The proof ends by a standard argument. We first show that $\Delta(n)$ is bounded. Denoting $p_n := \mathbb{E}[\mathbf{1}_{\{I_n=n\}}|A_n|^p]^{1/p}$ we have $p_n = o(1)$ by (35). Hence,

$$\Delta(n) \leq p_n\Delta(n) + \mathbb{E}[|A_n|^p] \max_{1\leq j\leq n-1} \Delta(j) + o(1). \qquad (46)$$

From (31) and (32) there exists $\xi = \|A\|_p^p < 1$ with $\|A_n\|_p^p = \xi + o(1)$. Hence we have

$$\Delta(n) \leq (\xi + o(1)) \max_{1\leq j\leq n-1} \Delta(j) + o(1). \qquad (47)$$

This implies that the $\Delta(n)$ remain bounded. Now, assume that $\limsup_{n\to\infty} \Delta(n) = c$ for some $c \geq 0$ and fix some $\varepsilon > 0$. Then there exists some $k \in \mathbb{N}$ such that $\Delta(n) \leq c + \varepsilon$ for all $n \geq k$ and by the technical condition (35), also some $N \geq k$ such that for all $n \geq N$

$$\mathbb{E}\big[|A_n|^p \mathbf{1}(I_n \leq k \vee I_n = n)\big] \leq \left(\frac{\varepsilon}{\sup_n \Delta(n)}\right)^p, \qquad (48)$$

$\|A_n\|_p \leq \|A\|_p + \varepsilon$ and the $o(1)$ in (45) is also bounded by ε. Thus, by (45),

$$\begin{aligned}
\Delta(n) &\leq \mathbb{E}[|A_n|^p \Delta(I_n)^p \mathbf{1}(I_n > k)]^{1/p} + \mathbb{E}[|A_n|^p \Delta(I_n)^p \mathbf{1}(I_n \leq k)]^{1/p} + \varepsilon \\
&\leq \mathbb{E}[|A_n|^p c^p]^{1/p} + \mathbb{E}[A_n^p \mathbf{1}(I_n \leq k)]^{1/p}\left(\sup_{m\in\mathbb{N}} \Delta(m)\right) + \varepsilon \\
&\leq (\|A\|_p + \varepsilon)(c + \varepsilon) + 2\varepsilon \qquad (49)
\end{aligned}$$

for $n \geq N$. Note that $\|A\|_p < 1$ because of (32). If c was bigger than 0, we could choose ε such that the right hand side of (49) is less than c, a contradiction to c being $\limsup_{n\to\infty} \Delta(n)$. Thus $c = \limsup_{n\to\infty} \Delta(n) = 0$, showing (37). ◀

5 Proof of Theorem 6 and Corollary 8

The proof of Theorem 6 is an application of Theorem 11 where the behaviour of Mergesort as stated in Lemma 3 and Lemma 4 is exploited.

Proof of Theorem 6. Let M_n denote the number of key comparisons Mergesort requires to sort a list of n uniformly permuted elements. We use M_n generically for both, Top-down or Queue-Mergesort. Since pivots are chosen independently from all other random sources (the permutation of the list and the cost of the Mergesort), for X_n, the number of key comparisons to sort a uniformly permuted list of n elements by QuickXsort with X being Mergesort, we have the distributional recurrence

$$X_n \stackrel{d}{=} X_{I_n} + M_{n-I_n-1} + n - 1, \qquad (50)$$

where I_n is the length of the sublist, where QuickXsort recurses after the first partitioning step. We have that on the right hand side of the latter display, I_n, $(X_0, \dots, X_{n-1})$ and $(M_0, \dots, M_{n-1})$ are independent. The distribution of I_n depends on the choice of the pivot element used by QuickXsort, we have that the distribution of I_n/n converges to a ϱ in ℓ_2, see (1). To normalize the X_n we introduce

$$\begin{aligned}
Y_n &:= \frac{X_n - n\log_2(n)}{n} && (51)\\
&\stackrel{d}{=} \frac{I_n}{n} \cdot \frac{X_{I_n} - I_n \log_2(I_n)}{I_n}\\
&\quad + \frac{n - I_n - 1}{n} \cdot \left(\frac{M_{n-I_n-1} - (n - I_n - 1)\log_2(n - I_n - 1)}{n - I_n - 1} - \Pi(\log_2(n - I_n - 1)) \right)\\
&\quad + \frac{n - I_n - 1}{n} \Pi(\log_2(n - I_n - 1))\\
&\quad + \frac{1}{n} \left(I_n \log_2(I_n) + (n - I_n - 1)\log_2(n - I_n - 1) - n\log_2(n) + n - 1 \right)\\
&=: A_n Y_{I_n} + b_n. && (52)
\end{aligned}$$

We aim for applying Theorem 11. We choose $d = 2$, Z having distribution ϱ, and $b^\star(x) := \mathcal{E}(Z) + (1 - Z)\Pi(x + \log_2(1 - Z))$ with Π being Π^{td} of Π^q depending on the Mergesort variant, see Lemma 3 and Lemma 4. Then, we have that (31) is satisfied since the summand containing M_{n-I_n-1} vanishes asymptotically due to the variance of Mergesort being of linear order. Also (32), (31) and (34) are fulfilled. We abbreviate $b_n^\star := b^\star(\log_2(n)) = \mathcal{E}(Z) + (1 - Z)\Pi(\log_2(n(1 - Z))$. The periodic function Π is bounded and continuous as is the function $x \to x\log(x) + (1 - x)\log(1 - x)$ on $[0, 1]$. Hence, $\sup_{n \in N} \|b_n\|_p < \infty$. Condition (35) is also fulfilled, as $I_n/n \le 1$, I_n tends to infinity and $\mathbb{P}(I_n = n) = 0$. Theorem 11 implies (12).

For n_k and x as in Theorem 11 we have

$$\ell_2\left(\frac{X_{n_k} - n_k \log_2 n_k}{n_k}, \Upsilon(x) \right) \le \ell_2(Y_{n_k}, \Upsilon(\{\log_2 n_k\})) + \ell_2(\Upsilon(\{x + \varepsilon_k\}), \Upsilon(x)). \tag{53}$$

Now, continuity and periodicity of $\Upsilon(\{x\})$ imply (13).

Note that ℓ_2 convergence implies convergence of the first and second moment. Hence, our nomalization in (51) and (12) imply (14). ◀

The functions Λ_E and Λ_V appearing in (14) can be computed from the recursion (36) or the sum (10):

$$\begin{aligned}
&\mathbb{E}[\Upsilon(x)]\\
&= \sum_{k=1}^{\infty} \mathbb{E}\left[\left(\prod_{j=1}^{k-1} Z_j \right) \left(\mathcal{E}(Z_k) + (1 - Z_k)\Pi\left(x + \log_2(1 - Z_k) + \sum_{j=1}^{k-1} \log_2(Z_j) \right) \right) \right]\\
&= \frac{\mathbb{E}[\mathcal{E}(Z)]}{1 - \mathbb{E}[Z]} + \sum_{k=1}^{\infty} \mathbb{E}\left[\left(\prod_{j=1}^{k-1} Z_j \right) (1 - Z_k)\Pi\left(x + \log_2(1 - Z_k) + \sum_{j=1}^{k-1} \log_2(Z_j) \right) \right]. && (54)
\end{aligned}$$

The last factor in the latter display can be bounded by taking the supremum over x, which implies:

$$\sup_{0 \le x \le 1} \mathbb{E}[\Upsilon(x)] \le \frac{\mathbb{E}[\mathcal{E}(Z)] + \mathbb{E}[1 - Z]\sup_{0 \le x \le 1} \Pi(x)}{1 - \mathbb{E}[Z]} \le \frac{\mathbb{E}[\mathcal{E}(Z)]}{1 - \mathbb{E}[Z]} + \sup_{0 \le x \le 1} \Pi(x). \tag{55}$$

The analogous bound applies to the infimum.

Proof of Corollary 8. To obtain bounds analogously to [8], we have to derive exact expressions for $\mathbb{E}[Z]$ and $\mathbb{E}[\mathcal{E}(Z)]$. In both Mergesort variants, we have $\alpha = 1/2$ and therefore Z has density $f_\varrho(x) = 2(\mathbf{1}_{(1/3,1/2)}(x) + \mathbf{1}_{(2/3,1)}(x))$ when using a single pivot. This yields us $\mathbb{E}[Z] = 25/36$ and

$$\mathbb{E}[\mathcal{E}(Z)] = 1 + \int_0^1 x\log_2(x) + (1-x)\log_2(1-x)\mathrm{d}x = 1 - \frac{1}{2\ln(2)},$$

where we used the symmetry of $\mathcal{E}(Z)$ in Z for the second calculation.

If we use the median-of-3 version, Z instead has density

$$f_\varrho(x) = 12x(1-x)(\mathbf{1}_{(1/3,1/2)}(x) + \mathbf{1}_{(2/3,1)}(x)),$$

which gives us $\mathbb{E}[Z] = \frac{263}{432}$ and

$$\mathbb{E}[\mathcal{E}(Z)] = 1 + 6\int_0^1 x^2(1-x)\log_2(x) + x(1-x)^2\log_2(1-x)\mathrm{d}x = 1 - \frac{7}{12\ln(2)}, \tag{56}$$

again using symmetry. In general: If we take $2h-1$ pivots for $h \geq 1$, we obtain $\mathbb{E}[\mathcal{E}(Z)] = 1 - \frac{H_{2h}-H_h}{\log(2)}$.

The function Π^{td} is bounded by -1.25265 ± 0.01185, see [8] and [14]. This implies

$$\mathbb{E}[\Upsilon(x)] = \frac{36}{11}\left(1 - \frac{1}{2\ln(2)}\right) - (1.25265 \pm 0.01185) = -0.3407 \pm 0.01185 \tag{57}$$

and therefore $\mathbb{E}[X_n] = n\log_2(n) - (0.3407 \pm 0.01185)n + o(n)$ for median-of-1. For median-of-3,

$$\mathbb{E}[\Upsilon(x)] = \frac{432}{169}\left(1 - \frac{7}{12\ln(2)}\right) - (1.25265 \pm 0.01185) = -0.84768 \pm 0.01185. \tag{58}$$

The same can be done for Queue-Mergesort using Π^q from (7). ◀

For $h = 1$, the function Λ_E can therefore be explicitly written as

$$\begin{aligned}\Lambda_E(x) =& \frac{36}{11}\left(1 - \frac{1}{2\ln(2)}\right)\\ &+ \sum_{k=1}^{\infty}\mathbb{E}\left[\left(\prod_{j=1}^{k-1} Z_j\right)(1-Z_k)\Pi\left(x + \log_2(1-Z_k) + \sum_{j=1}^{k-1}\log_2(Z_j)\right)\right].\end{aligned} \tag{59}$$

References

1 Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows in metric spaces and in the space of probability measures.* Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 2008.

2 Peter J. Bickel and David Freedman. Some asymptotic theory for the bootstrap. *Ann. Stat.*, 9:1196–1217, 1981. doi:10.1214/aos/1176345637.

3 Domenico Cantone and Gianluca Cincotti. QuickHeapsort, an efficient mix of classical sorting algorithms. *Theoretical Computer Science*, 285(1):25–42, 2002. Algorithms and Complexity. doi:10.1016/S0304-3975(01)00288-2.

4 Wei-Mei Chen, Hsien-Kuei Hwang, and Gen-Huey Chen. The cost distribution of Queue-Mergesort, optimal Mergesorts, and power-of-2 rules. *J. Algorithms*, 30(2):423–448, 1999. doi:10.1006/jagm.1998.0986.

5 Hua-Huai Chern, Michael Fuchs, Hsien-Kuei Hwang, and Ralph Neininger. Dependence and phase changes in random m-ary search trees. *Random Struct. Algorithms*, 50(3):353–379, 2017. doi:10.1002/rsa.20659.

6 Hua-Huai Chern and Hsien-Kuei Hwang. Phase changes in random m-ary search trees and generalized quicksort. *Random Struct. Algorithms*, 19(3-4):316–358, 2001. doi:10.1002/rsa.10005.

7 Stefan Edelkamp and Armin Weiß. QuickXsort: Efficient sorting with $n \log n - 1.399n + o(n)$ comparisons on average. In *Computer Science – Theory and Applications*, pages 139–152, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-06686-8_11.

8 Stefan Edelkamp, Armin Weiß, and Sebastian Wild. QuickXsort: A fast sorting scheme in theory and practice. *Algorithmica*, 82(3):509–588, March 2020. doi:10.1007/s00453-019-00634-0.

9 James Allen Fill and Nevin Kapur. The space requirement of m-ary search trees: distributional asymptotics for $m \geq 27$. Preprint, arXiv:math/0405144 [math.PR] (2004), 2004. arXiv:0405144.

10 Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, 2009.

11 Michael Fuchs, Hsien-Kuei Hwang, and Vytas Zacharovas. An analytic approach to the asymptotic variance of trie statistics and related structures. *Theoretical Computer Science*, 527:1–36, 2014. doi:10.1016/j.tcs.2014.01.024.

12 Mordecai J. Golin and Robert Sedgewick. Queue-mergesort. *Information Processing Letters*, 48(5):253–259, 1993. doi:10.1016/0020-0190(93)90088-Q.

13 Hsien-Kuei Hwang. Limit theorems for mergesort. *Random Struct. Algorithms*, 8(4):319–336, 1996. doi:10.1002/(SICI)1098-2418(199607)8:4<319::AID-RSA3>3.0.CO;2-0.

14 Hsien-Kuei Hwang. Asymptotic expansions of the mergesort recurrences. *Acta Inf.*, 35(11), 1998. doi:10.1007/s002360050147.

15 Jasper Ischebeck. Central limit theorems for additive functionals of patricia tries. *Journal of Applied Probability*, pages 1–14, 2025. doi:10.1017/jpr.2025.10034.

16 Svante Janson. Functional limit theorems for multitype branching processes and generalized Pólya urns. *Stochastic Processes Appl.*, 110(2):171–245, 2004. doi:10.1016/j.spa.2003.12.002.

17 Svante Janson. Central limit theorems for additive functionals and fringe trees in tries. *Electronic Journal of Probability*, 27(none):1–63, 2022. doi:10.1214/22-EJP776.

18 Svante Janson and Ralph Neininger. The size of random fragmentation trees. *Probab. Theory Relat. Fields*, 142(3-4):399–442, 2008. doi:10.1007/s00440-007-0110-1.

19 Donald E. Knuth. Von Neumann's first computer program. *ACM Comput. Surv.*, 2(4):247–260, December 1970. doi:10.1145/356580.356581.

20 Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Professional, Reading, MA, 2nd edition, 1998.

21 Péter Major. The approximation of partial sums of independent RV's. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 35(3):213–220, 1976.

22 Ralph Neininger. On a multivariate contraction method for random recursive structures with applications to quicksort. *Random Struct. Algorithms*, 19(3-4):498–524, 2001. doi:10.1002/rsa.10010.

23 Ralph Neininger and Ludger Rüschendorf. A general limit theorem for recursive algorithms and combinatorial structures. *Ann. Appl. Probab.*, 14(1):378–418, 2004. doi:10.1214/aoap/1075828056.

24 Svetlozar T. Rachev and Ludger Rüschendorf. *Mass transportation problems. Vol. I*. Probability and its Applications (New York). Springer-Verlag, New York, 1998. Theory.

25 Klaus Reinhardt. Sorting in-place with a worst case complexity of $n \log n - 1.3n + o(\log n)$ comparisons and $\epsilon n \log n + o(1)$ transports. In *Algorithms and Computation*, pages 489–498, Berlin, Heidelberg, 1992. Springer. doi:10.1007/3-540-56279-6_101.

26 Uwe Rösler. On the analysis of stochastic divide and conquer algorithms. *Algorithmica*, 29(1-2):238–261, 2001. doi:10.1007/BF02679621.

27 Uwe Rösler and Ludger Rüschendorf. The contraction method for recursive algorithms. *Algorithmica*, 29(1-2):3–33, 2001. doi:10.1007/BF02679611.

28 Wojciech Szpankowski. *Average case analysis of algorithms on sequences. With a foreword by Philippe Flajolet.* Wiley-Intersci. Ser. Discrete Math. Optim. Chichester: Wiley, 2001.

29 Cédric Villani. *Optimal transport*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 2009. doi:10.1007/978-3-540-71050-9.