Analysis of Quickselect under Yaroslavskiy's Dual-Pivoting Algorithm

Sebastian Wild* Markus E. Nebel* Hosam Mahmoud†

June 18, 2013

There is excitement within the algorithms community about a new partitioning method introduced by Yaroslavskiy. This algorithm renders Quicksort slightly faster than the case when it runs under classic partitioning methods. We show that this improved performance in Quicksort is *not* sustained in Quickselect; a variant of Quicksort for finding order statistics.

We investigate the number of comparisons made by Quickselect to find a key with a randomly selected rank under Yaroslavskiy's algorithm. This *grand averaging* is a smoothing operator over all individual distributions for specific fixed order statistics. We give the exact grand average. The grand distribution of the number of comparison (when suitably scaled) is given as the fixed-point solution of a distributional equation of a contraction in the Zolotarev metric space. Our investigation shows that Quickselect under older partitioning methods slightly outperforms Quickselect under Yaroslavskiy's algorithm, for an order statistic of a random rank. Similar results are obtained for extremal order statistics, where again we find the exact average, and the distribution for the number of comparisons (when suitably scaled). Both limiting distributions are of perpetuities (a sum of products of independent mixed continuous random variables).

AMS subject classifications: Primary: 60C05; secondary: 68P10, 68P20.

Keywords: sorting, Quicksort, Quickselect, combinatorial probability, algorithm, recurrence, average-case analysis, grand average, perpetuity, fixed point, metric, contraction.

1. Quicksort, What Is New?

Quicksort is a classic fast sorting algorithm. It was originally published by Hoare [17]; see also [6, 20, 22, 42]. Quicksort is the method of choice to implement a sorting function in many widely used program libraries, e. g. in the C/C++ standard library and in the Java runtime environment. The algorithm is recursive, and at each level of recursion it uses a partitioning algorithm. Classic implementations of Quicksort use a variety of partitioning algorithms derived from fundamental versions invented by Hoare [16, 17] and refined and popularized by Sedgewick [39, 40, 41, 43].

*Computer Science Department, University of Kaiserslautern, Germany {wild,nebel}@cs.uni-kl.de

†Hosam Mahmoud, Department of Statistics, The George Washington University, Washington, D.C. 20052, U.S.A.

Very recently, a partitioning algorithm proposed by Yaroslavskiy created some sensation in the algorithms community. The excitement arises from various indications, theoretical and experimental, that Quicksort on average runs faster under Yaroslavskiy's dual pivoting (see [46]). Indeed, after extensive experimentation Oracle adopted Yaroslavskiy's dual-pivot Quicksort as the default sorting method for their Java 7 runtime environment, a software platform used on 850 million computers worldwide.[1]

2. Classic Single-Pivot Quicksort and Quickselect

Quicksort is a two-sided algorithm for sorting data (also called *keys*). In a classic implementation, it puts a pivot key in its correct position, and arranges the data in two groups relative to that pivot. Keys smaller than the pivot are put in one group, the rest are placed in the other group. The two groups are then sorted recursively.

The one-sided version (Quickselect) of the algorithm can be used to find order statistics. This one-sided version of Quicksort is also known as Hoare's "Find" algorithm, which was first given in [16]. To find a certain order statistic, such as the first quartile, Quickselect goes through the partitioning stage, just as in Quicksort, then the algorithm decides whether the pivot is the sought order statistic or not. If it is, the algorithm terminates (announcing the pivot to be the sought element); if not, it recursively pursues only the group on the side where the order statistic resides. We know which side to choose, as the rank of the pivot becomes known after partitioning.

A standard measure for the analysis of a comparison-based sorting algorithm is the number of *data* comparisons it makes while sorting; see for example [19, 20]. Other types of comparison take place while sorting, such as index or pointer comparisons. However, they are negligible in view of the fact that they mostly occur at lower asymptotic orders, and any individual one of them typically costs considerably less than an individual data comparison. For instance, comparing two indices is a comparison of two short integers, while two keys can be rather long such as business records, polynomials, or DNA strands, typically each comprising tens or hundreds of thousands of nucleotides. So, these additional index and pointer comparisons are often ignored in the analysis. We shall follow this tradition.

It is worth mentioning that a fair contrast between comparison-based sorting algorithms and sorting algorithms based on other techniques (such as radix selection, which uses comparisons of bits) should resort to the use of one basis, such as how many bits are compared in both. Indeed, in comparing two very long bit strings, we can decide almost immediately that the two strings are different, if they differ in the first bit. In other instances, where the two strings are "similar," we may run a very long sequence of bit comparisons till we discover the difference. Attention to this type of contrast is taken up in [4, 9, 10, 11, 44]. Other associated cost measures include the number of swaps and data moves [23, 27].

3. Dual Pivoting

The idea of using two pivots (*dual-pivoting*) had been suggested before, see Sedgewick's and Hennequin's Ph. D. dissertations [15, 39]. Nonetheless, the implementations considered at the time did not show any promise. Analysis reveals that Sedgewick's dual-pivot Quicksort variant performs an asymptotic average of $\frac{32}{15} n \ln n + \mathcal{O}(n)$ data comparisons, while the classic (single-pivot) version uses only an asymptotic average of $2n \ln n + \mathcal{O}(n)$ comparisons [39, 46]. Hennequin's variant performs $2n \ln n + \mathcal{O}(n)$ comparisons [15] — asymptotically the same as classic Quicksort.

[1] estimate of Oracle, see `http://java.com/en/download/faq/whatis_java.xml`.

Algorithm 1 Dual-pivot Quickselect algorithm for finding the rth order statistic.

```
QUICKSELECT (A, left, right, r)
    // Assumes left ≤ r ≤ right.
    // Returns the element that would reside in A[r] after sorting A[left .. right].
 1  if right ≤ left
 2      return A[left]
 3  else
 4      (i_p, i_q) := PARTITIONYAROSLAVSKIY(A, left, right)
 5      c := sgn(r − i_p) + sgn(r − i_q)        // Here sgn denotes the signum function.
 6      case distinction on the value of c
 7          in case −2 do return QUICKSELECT (A, left , ℓ − 1, r)
 8          in case −1 do return A[i_p]
 9          in case  0 do return QUICKSELECT (A, ℓ + 1, g − 1, r)
10          in case +1 do return A[i_q]
11          in case +2 do return QUICKSELECT (A, g + 1, right, r)
12      end cases
13  end if
```

However, the inherently more complicated dual-pivot partitioning process is presumed to render it less efficient in practice.

These discoveries were perhaps a reason to discourage research on sorting with multiple-pivot partitioning, till Yaroslavskiy carefully heeded implementation details. His dual-partitioning algorithm improvement broke a psychological barrier. Would such improvements be sustained in Quickselect? It is our aim in this paper to answer this question. We find out that surprisingly there is no improvement in the number of comparisons: Quickselect under Yaroslavskiy's dual-pivot partitioning algorithm (simply Yaroslavskiy's algorithm, henceforth) is slightly *worse* than classic single-pivot Quickselect.

Suppose we intend to sort n distinct keys stored in the array $A[1\,..\,n]$. Dual partitioning uses *two* pivots, as opposed to the single pivot used in classic Quicksort. Let us assume the two pivots are initially $A[1]$ and $A[n]$, and suppose their ranks are p and q. If $p > q$, we swap the values of p and q. While seeking two positions for the two pivots, the rest of the data is categorized in three groups: small, medium and large. Small keys are those with ranks less than p, medium keys have ranks at least p and less than q, and large keys are those with ranks at least q. Small keys are moved to positions lower than p, large keys are moved to positions higher than q, medium keys are kept in positions between $p+1$ and $q-1$. So, the two keys with ranks p and q can be moved to their correct and final positions.

After this partitioning stage, dual-pivot Quicksort then invokes itself recursively (thrice) on $A[1\,..\,p-1]$, $A[p+1\,..\,q-1]$ and $A[q+1\,..\,n]$. The boundary conditions are the very small arrays of size 0 (no keys to sort), arrays of size 1 (such an array is already sorted), and arrays of size 2 (these arrays need only one comparison between the two keys in them); in these cases no further recursion is invoked.

This is the general paradigm for dual pivoting. However, it can be implemented in many different ways. Yaroslavskiy's algorithm keeps track of three pointers:

- ℓ, moving up from lower to higher indices, and below which all the keys have ranks less than p.

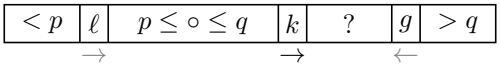
Algorithm 2 Yaroslavskiy's dual-pivot partitioning algorithm.

PartitionYaroslavskiy $(A, left, right)$

```
    // Assumes left ≤ right.
    // Rearranges A such that with (i_p, i_q) the return value holds { ∀ left ≤ j ≤ i_p    A[j] ≤ p
                                                                     { ∀ i_p ≤ j ≤ i_q    p ≤ A[j] ≤ q .
                                                                     { ∀ i_q ≤ j ≤ right  A[j] ≥ q
 1  if A[left] > A[right]
 2      p := A[right];        q := A[left]
 3  else
 4      p := A[left];         q := A[right]
 5  end if
 6  ℓ := left + 1;    g := right − 1;    k := ℓ
 7  while k ≤ g
 8      if A[k] < p
 9          Swap A[k] and A[ℓ]
10          ℓ := ℓ + 1
11      else
12          if A[k] ≥ q
13              while A[g] > q and k < g do g := g − 1 end while
14              if A[g] ≥ p
15                  Swap A[k] and A[g]
16              else
17                  Swap A[k] and A[g];   Swap A[k] and A[ℓ]
18                  ℓ := ℓ + 1
19              end if
20              g := g − 1
21          end if
22      end if
23      k := k + 1
24  end while
25  ℓ := ℓ − 1;      g := g + 1
26  A[left] := A[ℓ];          A[ℓ] := p       // Swap pivots to final positions
27  A[right] := A[g];         A[g] := q
28  return (ℓ, g)
```

- g, moving down from higher to lower indices, and above which all the keys have ranks at least q.
- k, moving up beyond ℓ and not past g. During the execution, all the keys at or below position k and above ℓ are medium, with ranks lying between p and q (both inclusively).

Hence, the three pointers ℓ, k and g divide the array into four ranges, where we keep the relation of elements invariantly as given above. Graphically, this reads as follows:



The adaptation of Quicksort to deliver a certain order statistic r (the rth smallest key) is straightforward. Once the positions for the two pivots are determined, we know whether $r < p$, $r = p$, $p < r < q$, $r = q$, or $r > q$. If $r = p$ or $r = q$, the algorithm declares one of the two pivots (now residing at position p or q) as the required rth order statistic, and terminates. If $r \neq p$ and $r \neq q$, the algorithm chooses one of three subarrays: If $r < p$ the algorithm recursively seeks the rth order statistic in $A[1\,..\,p-1]$, if $p < r < q$, the algorithm seeks the $(r-p)$th order statistic among the keys of $A[p+1\,..\,q-1]$; this $(r-p)$th key is, of course, ranked rth in the entire data set. If $r > q$, the algorithm seeks the $(r-q)$th order statistics in $A[q+1\,..\,n]$; this $(r-q)$th element is then ranked rth in the entire data set. Thus, only the subarray containing the desired order statistic is searched and the others are ignored.

Algorithm 1 is the formal algorithm in pseudo code. The code calls Yaroslavskiy's dual-pivot partitioning procedure (given as Algorithm 2). The code is written to work on the general subarray $A[\mathit{left}\,..\,\mathit{right}]$ in later stages of the recursion, and the initial call is QUICKSELECT$(A, 1, n, r)$.

Note that in Algorithm 2, variables p and q are used to denote the data elements used as pivots, whereas in the main text, p and q always refer to the *ranks* of these pivot elements relative to the current subarray. We kept the variables names in the algorithm to stay consistent with the literature.

Some words are in order to address the case of *equal elements*. If the input array contains equal keys, several competing notions of ranks exist. We choose an *ordinal ranking* that fits our situation best: The rank of an element is defined as its *index* in the array after sorting it with (a corresponding variant of) Quicksort. Consequently, Algorithm 1 correctly handles arrays with equal elements.

4. Probability Model

The standard probability model on data assumes the keys to be n real numbers independently sampled from a common *continuous* probability distribution — or equivalently, their ranks form a random permutation of $\{1, \ldots, n\}$ (all $n!$ permutations are equally likely); see [20].

4.1. Randomness Preservation

Several partitioning algorithms can be employed. A good partitioning algorithm produces subarrays following the random permutation model in subsequent recursive steps:

> *If the whole array is a (uniformly chosen) random permutation of its elements, so are the subarrays produced by partitioning.*

For instance, in classic single-pivot Quicksort, if p is the final position of the pivot, then right after the first partitioning stage the relative ranks of $A[1], \ldots, A[p-1]$ are a random permutation of $\{1, \ldots, p-1\}$ and the relative ranks of $A[p+1], \ldots, A[n]$ are a random permutation of $\{1, \ldots, n-p\}$, see [14] or [20].

Randomness preservation enhances performance on random data, and is instrumental in formulating recurrence equations for the analysis. Hoare's [17] and Lomuto's [3] single-pivot partitioning algorithms are known to enjoy this important and desirable property.

Lemma 1. *Yaroslavskiy's algorithm (Algorithm 2) is randomness preserving.*

Proof: Obviously, every key comparison in Yaroslavskiy's algorithm involves (at least) one pivot element; see lines 1, 8, 12, 13 and 14 of Algorithm 2. Hennequin shows that this is a sufficient criterion for randomness preservation [14], so Yaroslavskiy's algorithm indeed creates random subarrays. □

5. Main Results

We investigate the performance of Quickselect's number of data comparisons, when it seeks a key of a randomly selected rank, while employing Yaroslavskiy's algorithm. The exact grand average number of data comparisons is given in the following statement, in which H_n is the nth harmonic number $\sum_{k=1}^{n} 1/k$.

Proposition 1. *Let C_n be the number of data comparisons exercised while Quickselect is searching under Yaroslavskiy's algorithm for an order statistic chosen uniformly at random from all possible ranks. For $n \geq 4$,*

$$\mathbf{E}[C_n] \;=\; \tfrac{19}{6}n - \tfrac{37}{5}H_n + \tfrac{1183}{100} - \tfrac{37}{5}H_n n^{-1} - \tfrac{71}{300}n^{-1} \;\sim\; \tfrac{19}{6}n.$$

We use the notation $\stackrel{\mathcal{D}}{=}$ to mean (exact) equality in distribution, and $\xrightarrow{\mathcal{D}}$ to mean weak convergence in distribution.

Theorem 1. *Let C_n be the number of comparisons made by Quickselect with Yaroslavskiy's algorithm while searching for an order statistic chosen uniformly at random from all possible ranks. The random variables $C_n^* := C_n/n$ converge in distribution and in second moments to a limiting random variable C^* that satisfies the distributional equations*

$$\begin{aligned} C^* \;&\stackrel{\mathcal{D}}{=}\; U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\, C^* \;+\; \big(U_{(2)} - U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\, C^{*\prime} \\ &\qquad +\; \big(1 - U_{(2)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\, C^{*\prime\prime} \;+\; 1 + U_{(2)}\big(2 - U_{(1)} - U_{(2)}\big)\,. \end{aligned} \tag{1}$$

$$C^* \;\stackrel{\mathcal{D}}{=}\; X^* C^* + g(X^*, W^*)\,, \tag{2}$$

where $C^{*\prime}$ and $C^{*\prime\prime}$ are independent copies of C^*, which are also independent of $(U_{(1)}, U_{(2)}, V)$ and (X^*, W^*); $(U_{(1)}, U_{(2)})$ are the order statistics of two independent Uniform$(0,1)$ *random variables, V is a* Uniform$(0,1)$ *random variable independent of all else, and (X^*, W^*) have a bivariate density*

$$f(x,w) \;=\; \begin{cases} 6x, & \text{for } 0 < x < w < 1, \\ 0, & \text{elsewhere,} \end{cases}$$

and $g(X^*, W^*)$ is a fair mixture[2] of the three random variables

$$1 + W^*(2 - X^* - W^*), \quad 1 + (1 + X^* - W^*)(2W^* - X^*), \quad 1 + (1 - X^*)(X^* + W^*)\,.$$

[2] A fair mixture of three random variables is obtained by first choosing one of the three distributions at random, all three being equally likely, then generating a random variable from that distribution.

5. Main Results

As a corollary of Theorem 1, we find that $\mathbf{Var}[C_n^*] \sim \frac{25}{36}n^2 = 0.69\overline{4}n^2$, as $n \to \infty$. Another corollary is to write C^* explicitly as a sum of products of independent random variables:

$$C^* \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\infty} g(X_j, W_j)\Big(\prod_{k=1}^{j-1} X_j\Big), \qquad \text{where} \qquad X_j \stackrel{\mathcal{D}}{=} X^*, \qquad W_j \stackrel{\mathcal{D}}{=} W^*,$$

and $\{X_j\}_{j=1}^{\infty}$ is a family of totally independent random variables,[3] and so is $\{W_j^*\}_{j=1}^{\infty}$.

Remark *Let $\{V_j\}_{j=1}^{\infty}$ and $\{Z_k\}_{k=1}^{\infty}$ be two families of totally independent random variables, and assume V_j is independent of Z_k for each $j, k \geq 1$. Sums of products of independent random variables of the form*

$$V_1 + V_2Z_1 + V_3Z_1Z_2 + V_4Z_1Z_2Z_3 + \cdots$$

are called perpetuities. They appear in financial mathematics [8], in stochastic recursive algorithms [1], and in many other areas.

Proposition 2. *Let $\hat{C}_n$ be the number of comparisons made by Quickselect with Yaroslavskiy's algorithm to find the smallest key of a random input of size n. We then have*

$$\begin{aligned}\mathbf{E}\big[\hat{C}_n\big] &= \frac{1}{24n(n-1)(n-2)}\Big(57n^4 - 48n^3H_n - 178n^3 + 144n^2H_n \\ &\qquad + 135n^2 - 96nH_n - 14n + 24\Big), \qquad \text{for } n \geq 4, \\ &\sim \tfrac{19}{8}n\,.\end{aligned}$$

Theorem 2. *Let $\breve{C}_n$ be the number of comparisons made by Quickselect under Yaroslavskiy's algorithm on a random input of size n to find the rth order statistic, when $r = r_n = o(n)$ (i.e. small), and $n \to \infty$, and let $\bar{C}_n$ be its counterpart when the rank is large.*

The random variables $\breve{C}_n^* := \breve{C}_n/n$ and $\bar{C}_n^* := \bar{C}_n/n$ converge in distribution and in second moments to limiting random variables $\breve{C}^*$, and $\bar{C}^*$ satisfying the distributional equations

$$\breve{C}^* \stackrel{\mathcal{D}}{=} U_{(1)} \quad \breve{C}^* + 1 + U_{(2)}\big(2 - U_{(1)} - U_{(2)}\big), \tag{3}$$

$$\bar{C}^* \stackrel{\mathcal{D}}{=} \big(1 - U_{(2)}\big)\bar{C}^* + 1 + U_{(2)}\big(2 - U_{(1)} - U_{(2)}\big), \tag{4}$$

where $U_{(1)}$ and $U_{(2)}$ are respectively the minimum and maximum of two independent random variables, both distributed uniformly on $(0, 1)$. The limiting random variables $\breve{C}^*$ and $\bar{C}^*$ are distributed like perpetuities.

As a corollary, we find for $\breve{C}_n$ and $\bar{C}_n$, as $n \to \infty$,

$$\begin{aligned}\mathbf{E}[\breve{C}_n] &\sim \tfrac{19}{6}n, \quad \text{and} \quad \mathbf{Var}[\breve{C}_n] \sim \tfrac{1261}{4800}\,n^2 = 0.262708\overline{3}n^2\,, \\ \mathbf{E}[\bar{C}_n] &\sim \tfrac{19}{6}n, \quad \text{and} \quad \mathbf{Var}[\bar{C}_n] \sim \tfrac{1717}{4800}\,n^2 = 0.357708\overline{3}n^2\,.\end{aligned}$$

Another corollary is that $\breve{C}^*$ and $\bar{C}^*$ can be written explicitly as a sum of products of independent random variables:

$$\breve{C}^* \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\infty} Y_j\Big(\prod_{k=1}^{j-1} \breve{X}_j\Big), \quad \text{where} \quad \breve{X}_j \stackrel{\mathcal{D}}{=} U_{(1)}, \quad Y_j \stackrel{\mathcal{D}}{=} 1 + U_{(2)}\big(2 - U_{(1)} - U_{(2)}\big),$$

[3] For the usual definition of total independence see any classic book on probability, such as [5, p. 53], for example.

here $\{\breve{X}_j\}_{j=1}^{\infty}$ is a family of totally independent random variables, and so is $\{Y_j\}_{j=1}^{\infty}$. Similarly, we have

$$\bar{C}^* \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\infty} Y_j \Big(\prod_{k=1}^{j-1} \bar{X}_j \Big), \quad \text{where} \quad \bar{X}_j \stackrel{\mathcal{D}}{=} 1 - U_{(2)}, \quad Y_j \stackrel{\mathcal{D}}{=} 1 + U_{(2)} \big(2 - U_{(1)} - U_{(2)} \big),$$

where $\{\bar{X}_j\}_{j=1}^{\infty}$ is a another family of totally independent random variables.

6. Organization

The rest of the paper is devoted to the proof and is organized as follows. Section 7 sets up fundamental components of the analysis, and working notation that will be used throughout. In Section 8, we present a probabilistic analysis of Yaroslavskiy's algorithm. The analysis under rank smoothing is carried out in Section 9, which has two subsections: Subsection 9.1 is for the exact grand average, and Subsection 9.2 is for the limiting grand distribution via the contraction method. We say a few words in that subsection on the origin and recent developments of the method and its success in analyzing divide-and-conquer algorithms.

In Section 10, we import the methodology to obtain results for extremal order statistics. Again, Section 10 has two subsections: Subsection 10.1 is for the exact average, and Subsection 10.2 is for the limiting distribution. We conclude the paper in Section 11 with remarks on the overall perspective of the use of Yaroslavskiy's algorithm in Quicksort and Quickselect. The appendices are devoted to proving some technical points; Appendix A lays common foundations and Appendices B and C formally show convergence to limit law for random ranks, respectively extremal ranks.

7. Preliminaries and Notation

Let $C_n^{(r)}$ be the *number of comparisons* made by Quickselect under Yaroslavskiy's algorithm on a random input of size n to seek the rth order statistic. While this variable is easy to analyze for extremal values r (nearly smallest and nearly largest), it is harder to analyze this variable for intermediate values of r, such as when $r = \lfloor 0.17n \rfloor$. Typically, the analysis for median is hardest (when $r = \lfloor \frac{1}{2}n \rfloor$). It is also algorithmically hardest to find, see [38] for complexity bounds.

Analyzing $C_n^{(R)}$ when R itself is random provides smoothing over all possible values of $C_n^{(r)}$. So, we let $R = R_n$ be a random variable distributed like Uniform$[1\,..\,n]$. This rank randomization introduces a smoothing operation over the easy and hard cases that makes the problem of moderate complexity and amenable to analysis. In this case we can use the simplified notation $C_n := C_n^{(R_n)}$. One seeks a grand average of all averages, a grand variance of all variances and a grand (average) distribution of all distributions in the specific cases of r as a global measure over all possible order statistics. This smoothing technique was introduced in [24], and was used successfully in [21, 32]. Panholzer and Prodinger give a generating function formulation for grand averaging [31].

We shall use the following standard notation: $\mathbf{1}_{\mathcal{E}}$ is the *indicator random variable* of the event $\mathcal{E}$ that assumes the value 1, when $\mathcal{E}$ occurs, and assumes the value 0, otherwise. The notation $\xrightarrow{\mathcal{P}}$ stands for *convergence in probability*, and $\xrightarrow{a.s.}$ stands for *convergence almost surely*. By $\|X\|_p := \mathbf{E}[|X|^p]^{1/p}$, $1 \le p < \infty$, we denote the *L_p-norm* of random variable X, and we say random variables $X_1, X_2, \ldots$ *converge in L_p to X*, shortly written as $X_n \xrightarrow{L_p} X$, when $\lim_{n\to\infty} \|X_n - X\|_p = 0$. Unless otherwise stated, all asymptotic equivalents and bounds concern the limit as $n \to \infty$.

Let Hypergeo(n, s, w) be a *hypergeometric* random variable; that is the number of white balls in a size s sample of balls taken at random without replacement (all subsets of size s being equally likely) from an urn containing a total of n white and black balls, of which w are white.

Let P_n be the (random) *rank* of the smaller of the two pivots, and Q_n be the (random) rank of the larger of the two. For a random permutation, P_n and Q_n have a joint distribution uniform over all possible choices of a distinct pair of numbers from $\{1, \ldots, n\}$. That is to say,

$$\mathbf{Prob}(P_n = p, Q_n = q) \;=\; \frac{1}{\binom{n}{2}}, \qquad \text{for } 1 \le p < q \le n.$$

It then follows that P_n and Q_n have the marginal distributions

$$\begin{aligned} \mathbf{Prob}(P_n = p) &= \frac{n-p}{\binom{n}{2}}, \qquad \text{for } p = 1, \ldots, n-1\ ; \\ \mathbf{Prob}(Q_n = q) &= \frac{q-1}{\binom{n}{2}}, \qquad \text{for } q = 2, \ldots, n\ . \end{aligned}$$

Let $U_{(1)}$, and $U_{(2)}$ be the order statistics of U_1 and U_2, two independent continuous Uniform$(0, 1)$ random variables, i. e.

$$U_{(1)} \;=\; \min\{U_1, U_2\}\,, \qquad \text{and} \qquad U_{(2)} \;=\; \max\{U_1, U_2\}\ .$$

The two order statistics have the joint density

$$f_{(U_{(1)},U_{(2)})}(x, y) \;=\; \begin{cases} 2, & \text{for } 0 < x < y < 1; \\ 0, & \text{elsewhere,} \end{cases} \tag{5}$$

and consequently have the marginal densities

$$f_{U_{(1)}}(x) \;=\; \begin{cases} 2(1-x), & \text{for } 0 < x < 1; \\ 0, & \text{elsewhere,} \end{cases} \quad \text{and} \quad f_{U_{(2)}}(y) \;=\; \begin{cases} 2y, & \text{for } 0 < y < 1; \\ 0, & \text{elsewhere.} \end{cases} \tag{6}$$

The ensuing technical work requires all the variables to be defined on the same probability space. Let (U_1, U_2, V) be three independent Uniform$(0, 1)$ random variables defined on the probability space $([0, 1], \mathbb{B}_{[0,1]}, \lambda)$, where $\mathbb{B}_{[0,1]} = B \cap [0, 1]$, for B the usual *Borel sigma field* on the real line, and λ is the *Lebesgue measure*. We have

$$R_n \;\stackrel{\mathcal{D}}{=}\; \lceil nV \rceil \;\stackrel{\mathcal{D}}{=}\; \text{Uniform}[1\,..\,n]\ . \tag{7}$$

8. Analysis of Yaroslavskiy's Algorithm

An analysis of Quickselect using Yaroslavskiy's algorithm (Algorithm 2) requires a careful examination of this algorithm. In addition, being a novel partitioning method, it is a goal to analyze the method in its own right.

Let T_n be the number of comparisons exercised in the first call to Yaroslavskiy's algorithm. This quantity serves as a *toll function* for the recurrence relation underlying Quickselect: For unfolding the recurrence at size n, we have to pay a toll of T_n comparisons. The distribution of T_n below is given implicitly in the arguments of [46] and later used explicitly in [45, 47].

Lemma 2. *The number of comparisons T_n of Yaroslavskiy's partitioning method satisfies the following distributional equation conditional on (P_n, Q_n):*

$$T_n \overset{\mathcal{D}}{=} n-1 + \text{Hypergeo}(n-2, n-P_n-1, Q_n-2) + \text{Hypergeo}(n-2, \quad Q_n-2 \quad , n-Q_n) + 3\cdot \mathbf{1}_{\{A[Q_n]>\max\{A[1],A[n]\}\}}.$$

Corollary 1 ([46])**.** The expectation of T_n is given by

$$\mathbf{E}[T_n] = \tfrac{19}{12}(n+1) - 3.$$

Corollary 2 ([47])**.** The normalized number of comparisons $T_n^* := T_n \,/\, n$ converges to a limit T^* in L_2:

$$T_n^* \xrightarrow{L_2} T^*, \qquad \text{with} \qquad T^* \overset{\mathcal{D}}{=} 1 + U_{(2)}\big(2 - U_{(1)} - U_{(2)}\big)\,.$$

Remark 1. We re-obtain the leading term coefficient of $\mathbf{E}[T_n]$ as the mean of the limit distribution of T_n^*,

$$\mathbf{E}[T^*] = 1 + \int_0^1\int_0^y y(2-x-y) f_{(U_{(1)},U_{(2)})}(x,y)\,dx\,dy = \tfrac{19}{12}\,.$$

The bivariate density $f_{(U_{(1)},U_{(2)})}(x,y)$ is given in equation (5).

9. Analysis of Quickselect under Yaroslavskiy's Algorithm and Rank Smoothing

Right after the first round of partitioning, the two pivots (now moved to positions P_n and Q_n) split the data array into three subarrays: $A[1\,..\,P_n-1]$ containing keys with ranks smaller than P_n, $A[P_n+1\,..\,Q_n-1]$ containing keys with ranks between P_n and Q_n (both inclusively), and $A[Q_n+1\,..\,n]$ containing keys with ranks that are at least as large as Q_n. Quickselect is then invoked recursively on one of the three subarrays, depending on the desired order statistic.

As we have pairwise distinct elements almost surely, ranks are in one-to-one correspondence with key values and the three subarrays contain ranks *strictly* smaller, between and larger than the pivots. Therefore, we have the stochastic recurrence

$$C_n \overset{\mathcal{D}}{=} T_n + C_{P_n-1}\mathbf{1}_{\{R_n<P_n\}} + C'_{Q_n-P_n-1}\mathbf{1}_{\{P_n<R_n<Q_n\}} + C''_{n-Q_n}\mathbf{1}_{\{R_n>Q_n\}}, \tag{8}$$

where, for each $i \ge 0$, $C'_i \overset{\mathcal{D}}{=} C''_i \overset{\mathcal{D}}{=} C_i$, and $(C_{P_n}, C'_{Q_n-P_n-1}, C''_{n-Q_n})$ are conditionally independent (in the sense that, given $P_n = p$, and $Q_n = q$, C_{p-1}, C'_{q-p-1}, and C''_{n-p} are independent).

9.1. Exact Grand Average

The distributional equation (8) yields a recurrence for the average:

$$\mathbf{E}[C_n] = \mathbf{E}[T_n] + 3\,\mathbf{E}\big[C_{P_n-1}\mathbf{1}_{\{R_n<P_n\}}\big], \tag{9}$$

where symmetry is used to triple the term containing the first indicator. By conditioning on (P_n, Q_n) and the independent R_n, using Corollary 1 we get

$$\begin{aligned}\mathbf{E}[C_n] &= \mathbf{E}[T_n] + 3\sum_{1\le p<q\le n}\sum_{r=1}^{n}\mathbf{E}\big[C_{P_n-1}\mathbf{1}_{\{R_n<P_n\}} \,|\, P_n=p, Q_n=q, R_n=r\big] \\ &\qquad \times \mathbf{Prob}(P_n=p, Q_n=q, R_n=r) \\ &= \tfrac{19}{12}(n+1) - 3 + 3\sum_{p=1}^{n}\sum_{r=1}^{n}\mathbf{E}\big[C_{p-1}\mathbf{1}_{\{r<p\}}\big] \times \mathbf{Prob}(P_n=p)\,\mathbf{Prob}(R_n=r) \\ &= \tfrac{19}{12}(n+1) - 3 + \frac{6}{n^2(n-1)}\sum_{p=1}^{n}(p-1)(n-p)\,\mathbf{E}[C_{p-1}]. \qquad (10)\end{aligned}$$

This recurrence equation can be solved via generating functions. Let

$$A(z) := \sum_{n=0}^{\infty} n\,\mathbf{E}[C_n]\,z^n.$$

First, organize the recurrence (10) in the form

$$n^2(n-1)\,\mathbf{E}[C_n] = n^2(n-1)\big(\tfrac{19}{12}(n+1)-3\big) + 6\sum_{p=1}^{n}(p-1)(n-p)\,\mathbf{E}[C_{p-1}].$$

Next, multiply both sides of the equation by z^n (for $|z|<1$), and sum over $n\ge 3$, the range of validity of the recurrence, to get

$$z^2\sum_{n=3}^{\infty}\big(n^2(n-1)\,\mathbf{E}[C_n]\big)z^{n-2} = 6\sum_{n=3}^{\infty}\sum_{k=1}^{n}(n-k)(k-1)\,\mathbf{E}[C_{k-1}]z^n + g(z)\,,$$

where

$$g(z) = \sum_{n=3}^{\infty} n^2(n-1)\big(\tfrac{19}{12}(n+1)-3\big)\,z^n = \frac{z^3}{(1-z)^5}\big(7z^4-35z^3+70z^2-64z+60\big)\,.$$

Shifting summation indices and using the boundary conditions $C_0 = C_1 = 0$, and $C_2 = 1$, we extend the series to start at $n=0$, and get

$$z^2\big(A''(z) - 2^2(2-1)\cdot 1z^0\big) = 6\sum_{n=0}^{\infty}\sum_{k=0}^{n}(n-k)z^{n-k}\times\big(k\,\mathbf{E}[C_k]\big)z^k + g(z).$$

Finally, we get an Euler differential equation

$$z^2A''(z) = 6\frac{z^2}{(1-z)^2}A(z) + 4z^2 + g(z),$$

to be solved under the boundary conditions $A(0)=0$, and $A'(0)=0$. The solution to this differential equation is

$$\begin{aligned}A(z) = \frac{1}{300(1-z)^3}\Big(&2220z - 510z^2 + 830z^3 - 1185z^4 + 699z^5 - 154z^6 \\ &+ 2220(1-z)\ln(1-z)\Big).\end{aligned}$$

Extracting coefficients of z^n, we find for $n\ge 4$,

$$\mathbf{E}[C_n] = \tfrac{19}{6}n - \tfrac{37}{5}H_n + \tfrac{1183}{100} - \tfrac{37}{5n}H_n - \tfrac{71}{300n} \sim \tfrac{19}{6}n, \qquad \text{as } n\to\infty.$$

Proposition 1 is proved. □

9.2. Limit Distribution

Higher moments are harder to compute by direct recurrence as was done for the mean. For instance, exact variance computation involves rather complicated dependencies. We need a shortcut to determine the asymptotic distribution (i. e. all asymptotic moments), without resorting to exact calculation of each moment. A tool suitable for this task is the *contraction method*.

The contraction method was introduced by Rösler [34] in the analysis of the Quicksort algorithm, and it soon became a popular method because of the transparency it provides in the limit. Rachev and Rüschendorf added several useful extensions [33] and general contraction theorems, and multivariate extensions are available [29, 30, 35]. A valuable survey of the method was given by Rösler [37]. Neininger gives a variety of applications to random combinatorial structures and algorithms such as random search trees, random recursive trees, random digital trees and Mergesort [30]. The contraction method has also been used in the context of classic Quickselect [13, 36]. Other methods for the analysis of Quickselect have been used; for example, Grübel uses Markov chains [12].

We shall use the contraction method to find the grand distribution of Quickselect's number of comparisons under rank smoothing.

By dividing (8) by n and rewriting the fractions, we find

$$\begin{aligned}\frac{C_n}{n} \;\stackrel{\mathcal{D}}{=}\; & \frac{C_{P_n-1}}{P_n-1}\cdot\frac{P_n-1}{n}\mathbf{1}_{\{R_n<P_n\}} \\ & + \frac{C'_{Q_n-P_n-1}}{Q_n-P_n-1}\cdot\frac{Q_n-P_n-1}{n}\mathbf{1}_{\{P_n<R_n<Q_n\}} \\ & + \frac{C''_{n-Q_n}}{n-Q_n}\cdot\frac{n-Q_n}{n}\mathbf{1}_{\{R_n>Q_n\}} \\ & + \frac{T_n}{n}\,. \end{aligned} \tag{11}$$

This equation is conveniently expressed in terms of the *normalized* random variables $C_n^* := C_n/n$, that is

$$\begin{aligned}C_n^* \;\stackrel{\mathcal{D}}{=}\; & C^*_{P_n-1}\frac{P_n-1}{n}\mathbf{1}_{\{R_n<P_n\}} \;+\; C^{*\prime}_{Q_n-P_n-1}\frac{Q_n-P_n-1}{n}\mathbf{1}_{\{P_n<R_n<Q_n\}} \\ & +\; C^{*\prime\prime}_{n-Q_n}\frac{n-Q_n}{n}\mathbf{1}_{\{R_n>Q_n\}} \;+\; T_n^*,\end{aligned}$$

where for each $j \ge 0$, $C_j^{*\prime} \stackrel{\mathcal{D}}{=} C_j^{*\prime\prime} \stackrel{\mathcal{D}}{=} C_j^*$ and each of the families $\{C_j^*\}$, $\{C_j^{*\prime}\}$, $\{C_j^{*\prime\prime}\}$, $\{T_j\}$, and $\{R_j\}$ is comprised of totally independent random variables. This representation suggests a limiting functional equation as follows. If C_n^* converges to a limit C^*, so will $C^*_{P_n-1}$ because $P_n \to \infty$ almost surely, and it is plausible to guess that the combination $C^*_{P_n-1}\frac{P_n-1}{n}\mathbf{1}_{\{R_n<P_n\}}$ converges in distribution to $C^*U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}$. Likewise, it is plausible to guess that

$$C^{*\prime}_{Q_n-P_n-1}\frac{Q_n-P_n-1}{n}\mathbf{1}_{\{P_n<R_n<Q_n\}} \;\xrightarrow{\mathcal{D}}\; C^{*\prime}\big(U_{(2)}-U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\,,$$

and

$$C^{*\prime\prime}_{n-Q_n}\frac{n-Q_n}{n}\mathbf{1}_{\{R_n>Q_n\}} \;\xrightarrow{\mathcal{D}}\; C^{*\prime\prime}\big(1-U_{(2)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\,,$$

where $C^{*\prime} \stackrel{\mathcal{D}}{=} C^{*\prime\prime} \stackrel{\mathcal{D}}{=} C^*$, and $(C^*, C^{*\prime}, C^{*\prime\prime})$ are totally independent.

To summarize, if C_n^* converges in distribution to a limiting random variable C^*, one can guess that the limit satisfies the following distributional equation:

$$C^* \stackrel{\mathcal{D}}{=} U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\,C^* \;+\; \big(U_{(2)}-U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\,C^{*\prime} + \big(1-U_{(2)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\,C^{*\prime\prime} \;+\; T^*, \tag{12}$$

with $\big(C^*, C^{*\prime}, C^{*\prime\prime}, (U_{(1)}, U_{(2)}), V\big)$ being totally independent, and T^* as given in Corollary 2.

The formal proof of convergence is done by coupling the random variables C_n, P_n, Q_n, and R_n to be defined on the same probability space, and showing that the *distance* between the distributions of C_n^* and C^* converges to 0 in some suitable metric space of distribution functions. Here, we use the *Zolotarev metric* ζ_2, for which Neininger and Rüschendorf give convenient contraction theorems [30]. The technical details of using these theorems are provided in Appendix B. Finally, convergence in ζ_2 implies the claimed convergence in distribution and in second moments.

The representation in (12) admits direct calculation of asymptotic mean and variance. Taking expectations on both sides and exploiting symmetries gives

$$\begin{aligned}\mathbf{E}[C^*] &= \mathbf{E}\big[U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\big]\,\mathbf{E}[C^*] \;+\; \mathbf{E}\big[\big(U_{(2)}-U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\big]\,\mathbf{E}[C^{*\prime}] \\ &\qquad + \;\mathbf{E}\big[\big(1-U_{(1)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\big]\,\mathbf{E}[C^{*\prime\prime}] \;+\; \mathbf{E}[T^*] \\ &= 3\,\mathbf{E}\big[U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\big]\,\mathbf{E}[C^*] \;+\; \tfrac{19}{12}\,.\end{aligned}$$

So, we compute

$$\begin{aligned}\mathbf{E}\big[U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\big] &= \int_{x=0}^{1}\int_{v=0}^{1} x\,\mathbf{1}_{\{v<x\}}\,f_{U_{(1)}}(x)\,f_V(v)\,dv\,dx \\ &= 2\int_{x=0}^{1}\int_{v=0}^{x} x(1-x)\,dv\,dx \;=\; \tfrac{1}{6}\,.\end{aligned}$$

It follows that

$$\mathbf{E}[C^*] \;=\; \tfrac{19}{6}\,, \qquad \text{and, as } n\to\infty\,, \qquad \mathbf{E}[C_n] \;\sim\; \tfrac{19}{6}n\,.$$

Similarly, we can get the asymptotic variance of C_n. We only sketch this calculation. First, square the distributional equation (12), then take expectations. There will appear ten terms on the right-hand side. The three terms involving $(C^*)^2$ are symmetrical, and the three terms involving cross-products of indicators are 0 (the indicators are for mutually exclusive events). By independence, we have

$$\begin{aligned}\mathbf{E}\big[(C^*)^2\big] &= 3\,\mathbf{E}\big[U_{(1)}^2\mathbf{1}_{\{V<U_{(1)}\}}\big]\,\mathbf{E}\big[(C^*)^2\big] \\ &\quad + \;2\,\mathbf{E}\big[T^*\,U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\big]\,\mathbf{E}[C^*] \\ &\quad + \;2\,\mathbf{E}\big[T^*\big(U_{(2)}-U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\big]\,\mathbf{E}[C^{*\prime}] \\ &\quad + \;2\,\mathbf{E}\big[T^*\big(1-U_{(2)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\big]\,\mathbf{E}[C^{*\prime\prime}] \\ &\quad + \;\mathbf{E}\big[(T^*)^2\big].\end{aligned}$$

We show the computation for one of these ingredients:

$$\mathbf{E}\big[U_{(1)}^2\mathbf{1}_{\{V<U_{(1)}\}}\big] \;=\; 2\int_{y=0}^{1}\int_{x=0}^{y}\int_{v=0}^{1} x^2\mathbf{1}_{\{v<x\}}\,dv\,dx\,dy \;=\; \tfrac{1}{10}\,.$$

After carrying out similar calculations and using Corollary 2, we obtain

$$\mathbf{E}\big[(C^*)^2\big] \;=\; \tfrac{3}{10}\,\mathbf{E}\big[(C^*)^2\big] + 2\Big(\tfrac{43}{180}\,\mathbf{E}[C^*] + \tfrac{53}{180}\,\mathbf{E}[C^{*\prime}] + \tfrac{1}{4}\,\mathbf{E}[C^{*\prime\prime}]\Big) + \tfrac{229}{90}\,.$$

We can solve for $\mathbf{E}\big[(C^*)^2\big]$ and by inserting $\mathbf{E}[C^*] = \mathbf{E}[C^{*\prime}] = \mathbf{E}[C^{*\prime\prime}] = \frac{19}{6}$ get

$$\mathbf{E}\big[(C^*)^2\big] \;=\; \tfrac{10}{7}\Big(2\big(\tfrac{43}{180} + \tfrac{53}{180} + \tfrac{1}{4}\big)\tfrac{19}{6} + \tfrac{229}{90}\Big) \;=\; \tfrac{193}{18}\,.$$

The variance follows:

$$\begin{aligned}\mathbf{Var}[C_n] \;&=\; \mathbf{E}[C_n^2] - \big(\mathbf{E}[C_n]\big)^2 \;\sim\; \Big(\mathbf{E}[(C^*)^2] - \big(\mathbf{E}[C^*]\big)^2\Big)n^2\\ &=\; \big(\tfrac{193}{18} - \tfrac{361}{36}\big)n^2 \;=\; \tfrac{25}{36}n^2\,, \qquad \text{as } n\to\infty\,.\end{aligned}$$

We next present an explicit (unique) solution to the distributional equation (12). A random variable with this distribution takes the form of a perpetuity.

Lemma 3. *Every solution C^* of (12) also satisfies the distributional equation*

$$C^* \;\stackrel{\mathcal{D}}{=}\; X^*C^* + g(X^*, W^*)\,, \tag{13}$$

where the vector (X^*, W^*) is independent of C^*, and has bivariate density

$$f(x,w) = \begin{cases} 6x, & \text{for } \; 0 < x < w < 1;\\ 0, & \text{elsewhere;}\end{cases}$$

and $g(X^*, W^*)$ is a fair mixture of the three random variables

$$1 + W^*(2 - X^* - W^*), \quad 1 + (1 + X^* - W^*)(2W^* - X^*), \quad 1 + (1 - X^*)(X^* + W^*).$$

Proof: We show that the characteristic function $\phi_{C^*}(t)$ of variable C^* fulfilling equation (12) is also the characteristic function of a solution of (13). From (12), we find:

$$\begin{aligned}\phi_{C^*}(t) \;&=\; \mathbf{E}\Big[\exp\Big(it\Big(U_{(1)}\mathbf{1}_{\{V<U_{(1)}\}}\,C^* \;+\; \big(U_{(2)} - U_{(1)}\big)\mathbf{1}_{\{U_{(1)}<V<U_{(2)}\}}\\ &\qquad\qquad +\; \big(1 - U_{(2)}\big)\mathbf{1}_{\{V>U_{(2)}\}}\,C^{*\prime\prime} \;+\; T^*\Big)\Big)\Big]\\ &=\; \int_{y=0}^1\int_{x=0}^y\int_{v=0}^x \mathbf{E}\big[e^{it(xC^*+T^*)}\big]f_{(U_{(1)},U_{(2)},V)}(x,y,v)\,dv\,dx\,dy\\ &\quad + \int_{y=0}^1\int_{x=0}^y\int_{v=x}^y \mathbf{E}\big[e^{it((y-x)C^*+T^*)}\big]f_{(U_{(1)},U_{(2)},V)}(x,y,v)\,dv\,dx\,dy\\ &\quad + \int_{y=0}^1\int_{x=0}^y\int_{v=y}^1 \mathbf{E}\big[e^{it((1-y)C^*+T^*)}\big]f_{(U_{(1)},U_{(2)},V)}(x,y,v)\,dv\,dx\,dy\\ &=\; 2\int_{y=0}^1\int_{x=0}^y x\,\mathbf{E}\big[e^{it(xC^*+1+y(2-x-y))}\big]\,dx\,dy\\ &\quad + 2\int_{y=0}^1\int_{x=0}^y (y-x)\,\mathbf{E}\big[e^{it((y-x)C^*+1+y(2-x-y))}\big]\,dx\,dy\\ &\quad + 2\int_{y=0}^1\int_{x=0}^y (1-y)\,\mathbf{E}\big[e^{it((1-y)C^*+1+y(2-x-y))}\big]\,dx\,dy\,.\end{aligned}$$

In the middle integral, make the change of variables

$$x = 1 - u, \qquad y = 1 - u + v,$$

and in the rightmost integral make the change of variables

$$x = 1 - u, \qquad y = 1 - v,$$

to get the characteristic function in the form

$$\begin{aligned}
\phi_{C^*}(t) &= 2\int_{u=0}^{1}\int_{v=0}^{u} \phi_{C^*}(tv)\, v e^{it(1+u(2-v-u))}\, dv\, du \\
&\quad + 2\int_{u=0}^{1}\int_{v=0}^{u} \phi_{C^*}(tv)\, v e^{it(1+(1-u+v)(2u-v))}\, dv\, du \\
&\quad + 2\int_{u=0}^{1}\int_{v=0}^{u} \phi_{C^*}(tv)\, v e^{it(1+(1-v)(u+v))}\, dv\, du \\
&= \int_{w=0}^{1}\int_{x=0}^{w} \Big(\tfrac{1}{3}e^{it(1+w(2-x-w))} + \tfrac{1}{3}e^{it(1+(1+x-w)(2w-x))} + \tfrac{1}{3}e^{it(1+(1-w)(x+w))}\Big) \\
&\qquad \times (6x)\phi_{C^*}(tx)\, dx\, dw,
\end{aligned}$$

which is also the characteristic function of $X^*C^* + g(X^*, W^*)$. □

The representation in Lemma 3 allows us to obtain an expression for C^* as a sum of products of independent random variables. Toward this end, let $X_1, X_2, \ldots$ be independent copies of X^*, and let $Y_1, Y_2, \ldots$ be independent copies of $g(X^*, W^*)$, then

$$C^* \stackrel{\mathcal{D}}{=} Y_1 + X_1 C^* \stackrel{\mathcal{D}}{=} Y_1 + X_1(Y_2 + X_2 C^*).$$

Note that because C^* is independent of both X_1 and Y_1, the X and Y introduced in the iteration must be independent copies of X_1 and Y_1. Continuing the iterations (always introducing new independent random variables), we arrive at

$$\begin{aligned}
C^* &\stackrel{\mathcal{D}}{=} Y_1 + X_1 Y_2 + X_1 X_2 (Y_3 + X_3\, C^*) \\
&\;\;\vdots \\
&\stackrel{\mathcal{D}}{=} \sum_{j=1}^{M}\Big(Y_j \prod_{k=1}^{j-1} X_k\Big) + X_1 X_2 \cdots X_M\, C^*,
\end{aligned} \tag{14}$$

for any positive integer M. However, by the strong law of large numbers,

$$\frac{1}{M}\ln(X_1 X_2 \ldots X_M) \xrightarrow{a.s.} \mathbf{E}[\ln X^*] = -\tfrac{5}{6}, \qquad \text{as } M \to \infty\,,$$

and

$$X_1 X_2 \ldots X_M \xrightarrow{a.s.} 0, \qquad \text{as } M \to \infty\,.$$

Hence, we can proceed with the limit of (14) and write

$$C^* \stackrel{\mathcal{D}}{=} \sum_{j=1}^{\infty}\Big(Y_j \prod_{k=1}^{j-1} X_k\Big)\,.$$

10. Analysis of Quickselect under Yaroslavskiy's Algorithm for Extremal Ranks

The methods used for deriving results for dual-pivot Quickselect to locate a key of random rank carry over to the case of a relatively small or relatively large extremal order statistic (of order r_n, or $n - r_n$, for $r_n = o(n)$, as $n \to \infty$). We shall only sketch the argument and the result as they closely mimic what has been done for a random rank.

Let us first deal with the case of smallest rank. Let $\hat{C}_n := C_n^{(1)}$ be the number of key comparisons required by Quickselect running with Yaroslavskiy's algorithm to find the smallest element in an array $A[1\,..\,n]$ of random data. If the smaller of the two pivots (of random rank P_n) in the first round is not the smallest key, the algorithm always pursues the leftmost subarray $A[1\,..\,P_n - 1]$. We thus have the recurrence

$$\hat{C}_n \;\stackrel{\mathcal{D}}{=}\; \hat{C}_{P_n-1} + T_n\,. \tag{15}$$

10.1. Exact Mean

Equation (15) yields a recurrence for the average

$$\mathbf{E}[\hat{C}_n] \;=\; \mathbf{E}[T_n] + \mathbf{E}\big[\hat{C}_{P_n-1}\big].$$

Conditioning on P_n, we find

$$\begin{aligned}\mathbf{E}[C_n] \;&=\; \mathbf{E}[T_n] \;+\; \sum_{p=1}^{n} \mathbf{E}\big[C_{P_n-1} \,|\, P_n = p\big]\, \mathbf{Prob}(P_n = p)\\ &=\; \tfrac{19}{12}(n+1) - 3 \;+\; \frac{1}{\binom{n}{2}} \sum_{p=1}^{n} (n-p)\, \mathbf{E}[C_{p-1}]. \end{aligned}\tag{16}$$

This recurrence equation can be solved via generating functions, by steps very similar to what we did in Subsection 9.1, and we only give an outline of intermediate steps. If we let

$$\hat{A}(z) \;:=\; \sum_{n=0}^{\infty} \mathbf{E}[\hat{C}_n] z^n\,,$$

multiply (16) by $n(n-1)z^n$ and sum over $n \ge 3$, we get an Euler differential equation

$$z^2\big(\hat{A}''(z) - 2\big) \;=\; \frac{2z^2 \hat{A}(z)}{(1-z)^2} + h(z)\,,$$

with

$$h(z) \;:=\; \sum_{n=3}^{\infty} \big(\tfrac{19}{12}(n+1) - 3\big) n(n-1) z^n \;=\; \frac{z^3}{2(1-z)^4}\big(-7z^3 + 28z^2 - 42z + 40\big)\,.$$

This differential equation is to be solved under the boundary conditions $\hat{A}(0) = 0$, and $\hat{A}'(0) = 0$. The solution is

$$\hat{A}(z) \;=\; \frac{1}{24(1-z)^2}\big(36z + 42z^2 - 28z^3 + 7z^4 + 12\big(3 - 6z^2 + 4z^3 - z^4\big)\ln(1-z)\big)\,.$$

Extracting coefficients of z^n, we find, for $n \geq 4$,

$$\begin{aligned}\mathbf{E}[C_n^{(1)}] &= \frac{1}{24n(n-1)(n-2)}\Big(57n^4 - 48n^3 H_n - 178n^3 + 144n^2 H_n \\ &\qquad + 135n^2 - 96nH_n - 14n + 24\Big) \\ &\sim \tfrac{19}{8}n, \qquad \text{as } n \to \infty\,.\end{aligned}$$

10.2. Limit Distribution

By similar arguments as in case of random ranks we see that $\hat{C}_n^* := \hat{C}_n/n$ approaches $\hat{C}^*$, a random variable satisfying the distributions equation

$$\hat{C}^* \;\stackrel{\mathcal{D}}{=}\; U_{(1)}\,\hat{C}^* \;+\; T^*\,, \tag{17}$$

where $(U_{(1)}, T^*)$ is independent of $\hat{C}^*$. We formally establish convergence in law and second moments by showing that the distance of the distributions of $\hat{C}_n^*$ and $\hat{C}^*$ declines to 0 in the Zolotarev metric ζ_2. We go through the technical work using a handy theorem of Neininger and Rüschendorf [30] in Appendix C.

The representation in equation (17) allows us to obtain an expression for $\hat{C}^*$ by an unwinding process like that we used for C^*; one gets

$$\hat{C}^* \;\stackrel{\mathcal{D}}{=}\; \sum_{j=1}^{\infty}\Big(Y_j \prod_{k=1}^{j-1} X_k\Big)\,,$$

with $\{X_j\}_{j=1}^{\infty}$ and $\{Y_j\}_{j=1}^{\infty}$ being two families of totally independent random variables whose members are all distributed like $U_{(1)}$ respectively T^*.

The setup for any relatively small or relatively large order statistic (of rank r_n or $n - r_n$, for $r_n = o(n)$, as $n \to \infty$) is quite similar. Quickselect may not always choose the left side. For instance, for $r_n = o(n)$, the calls are typically to the left side, but occasionally the algorithm may invoke the recurrence on middle subarray or the one on the right-hand side. This happens when p or both p and q are too small, respectively, which are events with low probability for most recursive levels at the beginning. The stochastic recurrence is

$$C_n^{(r_n)} \;\stackrel{\mathcal{D}}{=}\; C_{P_n-1}^{(r_n)}\,\mathbf{1}_{\{r_n<P_n\}} \;+\; C'^{(r_n-P_n)}_{Q_n-P_n-1}\,\mathbf{1}_{\{P_n<r_n<Q_n\}} \;+\; C''^{(r_n-Q_n)}_{n-Q_n}\,\mathbf{1}_{\{r_n>Q_n\}} \;+\; T_n\,.$$

The two events $P_n < r_n < Q_n$ and $r_n > Q_n$ have ignorable probability of order $o(1)$. For large n, the recurrence is very approximately

$$C_n^{(r_n)} \;\stackrel{\mathcal{D}}{\approx}\; C_{P_n-1}^{(r_n)} + T_n\,,$$

where we used the symbol $\stackrel{\mathcal{D}}{\approx}$ to mean approximate equality in distribution. This is about the same as the recurrence for the smallest order statistic, and when all the technical work is done formally, one sees that $C_n^{(r_n)}/n$ satisfies the same limiting equation (17). Thus, $C_n^{(r_n)}/n$ and $C_n^{(1)}/n$ converge in distribution to the same limit $\hat{C}^*$, which has the representation given in (17). The argument for $C_n^{(n-r_n)}/n$ is symmetrical from the upper end and we get the corresponding limit. This completes a sketch of the proof of Theorem 2. □

Cost Measure		error	Quickselect with Yaroslavskiy's Algorithm	Classic Quickselect with Hoare's Algorithm
			when selecting a uniformly chosen order statistic	
Comparisons	expectation	$\mathcal{O}(\log n)$	$3.1\overline{6}n$	$3n$ †
	std. dev.	$o(n)$	$0.8\overline{3}n$	$1n$ †
Swaps	expectation	$\mathcal{O}(\log n)$	$1n$ ‡	$0.5n$ *
			when selecting an extremal order statistic	
Comparisons	expectation	$\mathcal{O}(\log n)$	$2.375n$	$2n$ †
	std. dev.	$o(n)$	small: $0.512551n$ large: $0.598087n$	$0.707107n$ †
Swaps	expectation	$\mathcal{O}(\log n)$	$0.75n$ ‡	$0.\overline{3}n$ *

† see [24]; Theorems 1 and 2.
‡ by the linear relation of expected swaps and comparisons, we can insert the toll function for swaps from [47].
* see [18]; integrating over $\alpha \in [0,1]$ resp. taking $\alpha \to 0$ in eq (12).

Table 1: Main results of this paper and the corresponding results for classic Quickselect from the literature.

11. Conclusion

In this paper, we discussed the prospect of running Quickselect making use of a dual-pivot partitioning strategy by Yaroslavskiy, which recently provided a speedup for Quicksort and is used today in the library sort of Oracle's Java 7. It has been proven that, for sorting, the total number of comparisons becomes smaller on average, upon using Yaroslavskiy's algorithm compared to the classic single-pivot variant [46]. Even if a single partitioning phase may need more comparisons than in the classic case, the reduced sizes of the subproblems to be processed recursively — the input to be sorted is partitioned into three instead of two parts — lead to an overall saving.

The speedup in Quicksort by Yaroslavskiy's partitioning algorithm, raise the hope for similar improvements in Quickselect. However, our detailed analysis, presented in this paper and summarized in Table 1, proves the opposite: When searching for a (uniformly) random rank, we find an expected number of comparisons of asymptotically $\frac{19}{6}n = 3.1\overline{6}n$ for a Quickselect variant running under Yaroslavskiy's algorithm, as opposed to $3n$ for classic Quickselect. For extremal cases, i. e. for ranks close to the minimum or maximum, an asymptotic average of $\frac{19}{8}n = 2.375n$ comparisons is needed, whereas the classic algorithm only uses $2n$. Though not considered here in detail, similar trends are observed for the number of swaps: In expectation, Quickselect under Yaroslavskiy's algorithm, also needs more swaps than the classic variant.

The observed inferiority of dual-pivoting in Quickselect goes beyond the case of Yaroslavskiy's algorithm. A simple argument shows that *any* dual-pivoting method must use at least $\frac{3}{2}n$ comparisons on average for a single partitioning step [2, Theorem 1]. Even with this optimal method, Quickselect needs $3n + o(n)$ comparisons on average to select a random order statistic and $2.25n + o(n)$ comparisons when searching for extremal elements; no improvement over classic Quickselect in both measures.

Our analysis provides deeper insight than just the average case — we derived variances and fixed-point equations for the distribution of the number of comparisons. Even though of less practical interest than the average considerations, it is worth noting that the variance of the number

of comparisons made by Quickselect under Yaroslavskiy's algorithm is significantly smaller than for the classic one, making actual costs more predictable.

We can give some intuition on why dual pivoting does not improve Quickselect based on our analysis. As already pointed out, the new algorithm may need more comparisons for a single partitioning round than the classic strategy, but leads to smaller subproblems to be handled recursively. In classic Quickselect, using pivot p and searching for a random rank r, we use n comparisons to exclude from recursive calls either $n-p$ elements, if $r < p$, or p elements, if $r > p$, or all n elements, if $r = p$. Averaging over all p and r, this implies that a single comparison helps to exclude $\frac{1}{3} + o(1)$ elements from further computations, as $n \to \infty$. When interpreting our findings accordingly, Quickselect under Yaroslavskiy's algorithm excludes asymptotically only $\frac{6}{19} \approx 0.3125$ elements per comparison on average. We have to conclude that the reduction in subproblem sizes is not sufficient to compensate for the higher partitioning costs.

Nevertheless, the attempts to improve the Quickselect algorithm are not a complete failure. Preliminary experimental studies give some hope for a faster algorithm in connection with cleverly chosen pivots (comparable to the median-of-three strategy well-known in the classic context), especially for presorted data. Future research may focus on this scenario, trying to identify an optimal choice for the pivots. Related results are known for classic Quickselect [26, 28] and Yaroslavskiy's algorithm in Quicksorting [48].

Furthermore, it would be interesting to extend our analysis to the number of bit comparisons instead of atomic key comparisons. This is especially of interest in connection with nonprimitive data types like strings. However, in this context one typically has to deal with much more complicated analysis for the resulting subproblems no longer preserve randomness in the subarrays (see [11] for corresponding results for classic Quickselect). As a consequence, the methods used in this paper are no longer applicable.

References

[1] Gerold Alsmeyer, Alex Iksanov, and Uwe Rösler. On distributional properties of perpetuities. *J. Theor. Probab.*, 22:666–682, 2009.

[2] Martin Aumüller and Martin Dietzfelbinger. Optimal Partitioning for Dual Pivot Quicksort. March 2013.

[3] Jon Bentley. Programming pearls: how to sort. *Commun. ACM*, 27(4):287–291, April 1984.

[4] Patrick Bindjeme and James Fill. The limiting distribution for the number of symbol comparisons used by quicksort is nondegenerate (extended abstract). *DMTCS Proceedings*, 0(01), 2012.

[5] Kai Lai Chung. *A Course in Probability Theory.* Academic Press, 3rd edition, 2001.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, 3rd edition, 2009.

[7] Herbert A. David and Haikady N. Nagaraja. *Order Statistics (Wiley Series in Probability and Statistics).* Wiley-Interscience, 3rd edition, 2003.

[8] Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling Extremal Events.* Springer Verlag, Berlin, Heidelberg, 1997.

[9] James Fill and Svante Janson. The number of bit comparisons used by Quicksort: an average-case analysis. *Electron. J. Probab.*, 17:no. 43, 1–22, 2012.

[10] James Allen Fill. Distributional convergence for the number of symbol comparisons used by QuickSort. *Ann. Appl. Probab.*, 23(3):1129–1147, June 2013.

[11] James Allen Fill and Takéhiko Nakama. Analysis of the Expected Number of Bit Comparisons Required by Quickselect. *Algorithmica*, 58(3):730–769, March 2009.

[12] Rudolf Grübel. Hoare's Selection Algorithm: A Markov Chain Approach. *J. Appl. Probab.*, 35(1):36–45, 1998.

[13] Rudolf Grübel and Uwe Rösler. Asymptotic Distribution Theory for Hoare's Selection Algorithm. *Adv. Appl. Probab.*, 28(1):252–269, 1996.

[14] Pascal Hennequin. Combinatorial analysis of Quicksort algorithm. *Rairo-Inf. Theor. Appl.*, 23(3):317–333, 1989.

[15] Pascal Hennequin. *Analyse en moyenne d'algorithmes : tri rapide et arbres de recherche.* PhD Thesis, Ecole Politechnique, Palaiseau, 1991.

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

[17] C. A. R. Hoare. Quicksort. *Comput. J.*, 5(1):10–16, January 1962.

[18] Hsien-Kuei Hwang and Tsung-Hsi Tsai. Quickselect and Dickman function. *Comb. Probab. Comput.*, 11, 2000.

[19] Peter Kirschenhofer and Helmut Prodinger. Comparisons in Hoare's Find Algorithm. *Comb. Probab. Comput.*, 7(01):111–120, 1998.

[20] Donald E. Knuth. *The Art Of Computer Programming: Searching and Sorting.* Addison Wesley, 2nd edition, 1998.

[21] Janice Lent and Hosam M. Mahmoud. Average-case analysis of multiple Quickselect: An algorithm for finding order statistics. *Stat. & Probab. Lett.*, 28(4):299–310, 1996.

[22] Hosam M. Mahmoud. *Sorting: A distribution theory.* John Wiley & Sons, Hoboken, NJ, USA, 2000.

[23] Hosam M. Mahmoud. Distributional analysis of swaps in Quick Select. *Theor. Comput. Sci.*, 411:1763–1769, 2010.

[24] Hosam M. Mahmoud, Reza Modarres, and Robert T. Smythe. Analysis of quickselect : an algorithm for order statistics. *Rairo-Inf. Theor. Appl.*, 29(4):255–276, 1995.

[25] Hosam M. Mahmoud and Boris Pittel. On the joint distribution of the insertion path length and the number of comparisons in search trees. *Discret. Appl. Math.*, 20(3):243–251, July 1988.

[26] Conrado Martínez, Daniel Panario, and Alfredo Viola. Adaptive sampling strategies for quickselects. *ACM Trans. Algorithm.*, 6(3):1–45, June 2010.

[27] Conrado Martínez and Helmut Prodinger. Moves and displacements of particular elements in Quicksort. *Theor. Comput. Sci.*, 410(21–23):2279–2284, 2009.

[28] Conrado Martínez and Salvador Roura. Optimal Sampling Strategies in Quicksort and Quickselect. *SIAM J. Comput.*, 31(3):683, 2001.

[29] Ralph Neininger. On a multivariate contraction method for random recursive structures with applications to Quicksort. *Random Struct. & Algorithm.*, 19(3-4):498–524, 2001.

[30] Ralph Neininger and Ludger Rüschendorf. A General Limit Theorem for Recursive Algorithms and Combinatorial Structures. *Ann. Appl. Probab.*, 14(1):378–418, 2004.

[31] Alois Panholzer and Helmut Prodinger. A generating functions approach for the analysis of grand averages for multiple QUICKSELECT. *Random Struct. & Algorithm.*, 13(3-4):189–209, 1998.

[32] Helmut Prodinger. Multiple Quickselect—Hoare's Find algorithm for several elements. *Inf. Process. Lett.*, 56(3):123–129, 1995.

[33] Svetlozar T. Rachev and Ludger Rüschendorf. Probability Metrics and Recursive Algorithms. *Adv. Appl. Probab.*, 27(3):770–799, 1995.

[34] Uwe Rösler. A limit theorem for "quicksort". *Rairo-Inf. Theor. Appl.*, 25(1):85–100, 1991.

[35] Uwe Rösler. On the analysis of stochastic divide and conquer algorithms. *Algorithmica*, 29(1):238–261, 2001.

[36] Uwe Rösler. QUICKSELECT revisited. *J. Iran. Stat. Inst.*, 3:271–296, 2004.

[37] Uwe Rösler and Ludger Rüschendorf. The contraction method for recursive algorithms. *Algorithmica*, 29(1):3–33, 2001.

[38] Arnold Schönhage, Mike Paterson, and Nicholas Pippenger. Finding the median. *J. Comput. & Syst. Sci.*, 13(2):184–199, October 1976.

[39] Robert Sedgewick. *Quicksort*. PhD Thesis, Stanford University, 1975.

[40] Robert Sedgewick. The analysis of Quicksort programs. *Acta Inf.*, 7(4):327–355, 1977.

[41] Robert Sedgewick. Implementing Quicksort programs. *Commun. ACM*, 21(10):847–857, October 1978.

[42] Robert Sedgewick and Philippe Flajolet. *An Introduction to the Analysis of Algorithms*. Addison-Wesley-Longman, 1996.

[43] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley, 4th edition, 2011.

[44] Brigitte Vallée, Julien Clément, James Allen Fill, and Philippe Flajolet. The Number of Symbol Comparisons in QuickSort and QuickSelect. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris Nikoletseas, and Wolfgang Thomas, editors, *ICALP 2009*, volume 5555 of *LNCS*, pages 750–763, Berlin, Heidelberg, 2009. Springer Verlag.

[45] Sebastian Wild. Java 7's Dual Pivot Quicksort. Master thesis, University of Kaiserslautern, 2012.

[46] Sebastian Wild and Markus E. Nebel. Average Case Analysis of Java 7's Dual Pivot Quicksort. In Leah Epstein and Paolo Ferragina, editors, *ESA 2012*, volume 7501 of *LNCS*, pages 825–836. Springer Berlin/Heidelberg, 2012.

[47] Sebastian Wild, Markus E. Nebel, and Ralph Neininger. Average Case and Distributional Analysis of Java 7's Dual Pivot Quicksort. *ACM Trans. Algorithm.*, submitted.

[48] Sebastian Wild, Markus E. Nebel, Raphael Reitzig, and Ulrich Laube. Engineering Java 7's Dual Pivot Quicksort Using MaLiJAn. In Peter Sanders and Norbert Zeh, editors, *ALENEX 2013*, pages 55–69. SIAM, 2013.

Appendix

A. Spacings and Subproblem Sizes

The only data aspect that plays any role in a comparison-based sorting or selection method is the *relative ranking*. For example, comparing 80 to 60 is the same as comparing 2 to 1 (and the same action is taken in both cases, like a swapping, for instance). All probability models that give a random permutation of ranks almost surely are therefore equivalent from the point of view of a comparison-based sorting method. For example, sorting data from *any* continuous distribution gives rise to the same distribution of complexity measures, such as the number of comparisons or swaps. We might as well work through the analysis using one convenient continuous data model, like the Uniform$(0,1)$. In this appendix, we assume our data to come from such a uniform density, as was done in [25, 47], which gives a convenient definition of random permutations of increasing sample sizes (and infinite random permutations, as well).

We use a notation that symmetrizes the reference to the three subarrays (see [47]). Instead of referring to the sizes of the three subarrays by $P_n - 1$, $Q_n - P_n - 1$, and $n - Q_n$, we simply call them $I_j^{(n)}$, for $j = 1, 2, 3$. Moreover, P_n and Q_n are an inconvenient basis for the transition to the limit as they take values from the discrete set $\{1, \ldots, n\}$; hence we use an alternative description of the distribution of $\mathbf{I}^{(n)} = (I_1^{(n)}, I_2^{(n)}, I_3^{(n)})$:

Denote by $\mathbf{S} = (S_1, S_2, S_3)$ the *spacings* induced by the two independent random variables U_1 and U_2 distributed uniformly in $(0, 1)$; formally we have

$$(S_1, S_2, S_3) \;=\; (U_{(1)},\, U_{(2)} - U_{(1)},\, 1 - U_{(2)})\,,$$

for

$$U_{(1)} := \min\{U_1, U_2\}, \qquad \text{and} \qquad U_{(2)} := \max\{U_1, U_2\}\,.$$

It is well-known that $\mathbf{S}$ is uniformly distributed in the standard 2-simplex [7, p. 133f], i. e. (S_1, S_2) has density

$$f_S(x_1, x_2) \;=\; \begin{cases} 2, & \text{for } x_1, x_2 \geq 0 \;\wedge\; x_1 + x_2 \leq 1, \\ 0, & \text{otherwise}\,. \end{cases}$$

$S_3 = 1 - S_1 - S_2$ is fully determined by (S_1, S_2).

We can express the distribution of the sizes of the three subarrays $\mathbf{I}^{(n)} = (I_1^{(n)}, I_2^{(n)}, I_3^{(n)})$ generated in the first partitioning round based on $\mathbf{S}$: We have $I_1^{(n)} + I_2^{(n)} + I_3^{(n)} = n - 2$, and conditional on $\mathbf{S}$, the vector $\mathbf{I}^{(n)}$ has a multinomial distribution:

$$\mathbf{I}^{(n)} \;\stackrel{\mathcal{D}}{=}\; \text{Mult}(n - 2;\, S_1, S_2, S_3).$$

Lemma 4. *We have for $j \in \{1, 2, 3\}$ the convergence, as $n \to \infty$,*

$$\frac{I_j^{(n)}}{n} \;\xrightarrow{L_2}\; S_j\,.$$

Proof: By the *strong law of large numbers*, $I_j^{(n)}/n$ converges to S_j almost surely. Moreover, $|I_j^{(n)}/n|$ is bounded by 1, and the statement follows. □

B. Proof of Convergence: Rank Smoothing

We are going to apply the following theorem by Neininger and Rüschendorf [30], which we restate for the reader's convenience:

Theorem B.1 ([30, Theorem 4.1]). *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of s-integrable, $0 < s \le 3$, random variables satisfying the recurrence*

$$X_n \stackrel{\mathcal{D}}{=} \sum_{r=1}^{K} A_r^{(n)} X_{I_r^{(n)}}^{(r)} + b^{(n)}, \qquad n \ge n_1,$$

where $K \in \mathbb{N}$ is a constant, $(\mathbf{A}^{(n)}, b^{(n)}, \mathbf{I}^{(n)})$ and $(X_n^{(1)})_{n\in\mathbb{N}}$, …, $(X_n^{(K)})_{n\in\mathbb{N}}$ are independent and $X_i^{(r)}$ is distributed like X_i for all $r = 1, \dots, K$ and $i \ge 0$. Assume further for all n: (a) if $0 < s \le 1$ that X_n has finite variance; (b) if $1 < s \le 2$ additionally that $\mathbf{E}[X_n] = 0$ and (c) if $2 < s \le 3$ additionally that $\mathbf{Var}[X_n] = 1$. Moreover, assume the following conditions:

(A) $(A_1^{(n)}, \dots, A_K^{(n)}, b^{(n)}) \xrightarrow{L_s} (A_1^*, \dots, A_K^*, b^*)$;

(B) $\mathbf{E}\big[\sum_{r=1}^{K} |A_r^*|^s\big] < 1$;

(C) For all $c \in \mathbb{N}$ and $r = 1, \dots, K$ holds $\mathbf{E}\big[\mathbf{1}_{\{I_r^{(n)} \le c \vee I_r^{(n)} = n\}} |A_r^{(n)}|^s\big] \to 0$.

Then $\lim_{n\to\infty} \zeta_s(X_n, X) = 0$ *for a random variable X whose distribution is given as the unique fixed point of*

$$X \stackrel{\mathcal{D}}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*$$

among all distributions with (a) finite sth moments, if $0 < s \le 1$, (b) finite sth moments and mean zero, if $1 < s \le 2$, and (c) finite sth moments, variance one and mean zero, if $2 < s \le 3$. Here, $(A_1^*, \dots, A_K^*, b^*)$ and $X^{(1)}$, …, $X^{(K)}$ are independent and $X^{(r)}$ is distributed like X for $r = 1, \dots, K$. □

See Section 2 of [30] for definition and discussion of the Zolotarev metrics ζ_s. For the application in this paper, it suffices to restate the following properties where we write $X_n \xrightarrow{\zeta_s} X$ to mean $\zeta_s(X_n, X) \to 0$:

(I) $X_n \xrightarrow{\zeta_s} X \implies X_n \xrightarrow{\mathcal{D}} X$, see [30, p. 382].

(II) $X_n \xrightarrow{\zeta_1} X \implies \mathbf{E}[X_n] \to \mathbf{E}[X]$, see [30, Remark p. 398].

(III) $X_n \xrightarrow{\zeta_2} X \implies \mathbf{Var}[X_n] \to \mathbf{Var}[X]$, see [30, Remark p. 398].

To prove convergence in distribution and in second moments of the number of comparisons used by Yaroslavskiy's algorithm under rank smoothing, we apply Theorem B.1 with $K = 3$ and $s = 2$. For $s = 2$, Theorem B.1 requires *centered* random variables — so we cannot directly use C_n^*. Guessing that $\mathbf{Var}[C_n] = \Theta(n^2)$, we define

$$C_n^\circ := \frac{C_n - \mathbf{E}[C_n]}{n}.$$

(Note the difference between C_n^* and C_n°.) Whence, $\mathbf{E}[C_n]$ is given in Proposition 1. Equation (11) can be written in terms of C_n° by subtracting $\mathbf{E}[C_n]$ on both sides and rewriting the right hand side as follows:

$$C_n^\circ \stackrel{\mathcal{D}}{=} C_{I_1}^\circ \frac{I_1^{(n)}}{n} \mathbf{1}_{\mathcal{E}_1^{(n)}} + C_{I_2}^{\circ\prime} \frac{I_2^{(n)}}{n} \mathbf{1}_{\mathcal{E}_2^{(n)}} + C_{I_3}^{\circ\prime\prime} \frac{I_3^{(n)}}{n} \mathbf{1}_{\mathcal{E}_3^{(n)}} + \frac{1}{n}\Big(T_n - \mathbf{E}[C_n] + \sum_{j=1}^{3} \mathbf{1}_{\mathcal{E}_j^{(n)}} \mathbf{E}\big[C_{I_j} \,|\, I_j^{(n)}\big]\Big).$$

Here $C_n^{\circ\prime}$ and $C_n^{\circ\prime\prime}$ are independent copies of C_n° and $\mathcal{E}_j^{(n)}$ is the event that the search continues in subproblem j (for $j = 1, 2, 3$):

$$\mathcal{E}_1^{(n)} := \{R_n < P_n\}, \qquad \mathcal{E}_2^{(n)} := \{P_n < R_n < Q_n\}, \qquad \mathcal{E}_3^{(n)} := \{Q_n < R_n\}.$$

B. Proof of Convergence: Rank Smoothing

In the proof that follows, we express these events via the distributional equation (7)

$$\mathcal{E}_1^{(n)} \stackrel{\mathcal{D}}{=} \{V < \tfrac{I_1^{(n)}}{n}\}, \qquad \mathcal{E}_2^{(n)} \stackrel{\mathcal{D}}{=} \{\tfrac{I_1^{(n)}+1}{n} < V < \tfrac{I_1+I_2+1}{n}\}, \qquad \mathcal{E}_3^{(n)} \stackrel{\mathcal{D}}{=} \{\tfrac{I_1^{(n)}+I_2^{(n)}+2}{n} < V\}.$$

Now, we show that the three conditions (A), (B) and (C) of Theorem B.1 are fulfilled:

Cond. (A) We first consider the coefficients $A_j^{(n)}$. For $j = 1, 2, 3$, we have

$$A_j^{(n)} = \tfrac{I_j^{(n)}}{n}\mathbf{1}_{\mathcal{E}_j^{(n)}} \xrightarrow{L_2} S_j\mathbf{1}_{\mathcal{E}_j},$$

where the limiting events $\mathcal{E}_j$ are defined as

$$\mathcal{E}_1 := \{V < S_1\}, \qquad \mathcal{E}_2 := \{S_1 < V < S_1 + S_2\}, \qquad \mathcal{E}_3 := \{S_1 + S_2 < V\}.$$

It is essential that $\mathcal{E}_j^{(n)}$ and $\mathcal{E}_j$ are defined in terms of the *same* random variable V; this couples the events and allows us to show the above convergence. The proof is a standard, but somewhat tedious computation:

For $j = 1$, we condition on $\mathbf{I}$ and $\mathbf{S}$ to expand the indicator variables:

$$\begin{aligned}
\mathbf{E}\Big[\Big(\tfrac{I_1^{(n)}}{n}\mathbf{1}_{\mathcal{E}_1^{(n)}} - S_1\mathbf{1}_{\mathcal{E}_1}\Big)^2\Big] &\le \mathbf{E}\Big[\mathbf{Prob}\Big(V < \min\big\{\tfrac{I_1^{(n)}}{n}, S_1\big\}\Big)\cdot\Big(\tfrac{I_1^{(n)}}{n} - S_1\Big)^2 \\
&\quad + \mathbf{Prob}\Big(\min\big\{\tfrac{I_1^{(n)}}{n}, S_1\big\} < V < \max\big\{\tfrac{I_1^{(n)}}{n}, S_1\big\}\Big)\cdot\max^2\big\{\tfrac{I_1^{(n)}}{n}, S_1\big\}\Big] \\
&\le \mathbf{E}\Big[\Big(\tfrac{I_1^{(n)}}{n} - S_1\Big)^2 + \Big|\tfrac{I_1^{(n)}}{n} - S_1\Big|\Big] \\
&= \Big\|\tfrac{I_1^{(n)}}{n} - S_1\Big\|_2 + \Big\|\tfrac{I_1^{(n)}}{n} - S_1\Big\|_1 \\
&\to 0, \qquad \text{for } n \to \infty,
\end{aligned}$$

where the second inequality uses that the factors are bounded by 1 uniformly in n and the last step follows by Lemma 4.

The convergence of the terms for $j = 3$ are very much the same by symmetry. In the case for $j = 2$, we have some more cases to distinguish, as the corresponding events $\mathcal{E}_2^{(n)}$ and $\mathcal{E}_2$ contain upper *and* lower bounds. We skip details similar in nature to the case $j = 1$.

The second part of condition (A) concerns the convergence of the toll function. We show:

$$\frac{1}{n}\Big(T_n - \mathbf{E}[C_n] + \sum_{j=1}^{3}\mathbf{1}_{\mathcal{E}_j^{(n)}}\mathbf{E}\big[C_{I_j^{(n)}} \mid I_j^{(n)}\big]\Big) \xrightarrow{L_2} T^* + \tfrac{19}{6}\Big(-1 + \sum_{j=1}^{3} S_j\mathbf{1}_{\mathcal{E}_j}\Big). \tag{18}$$

As $X_n \xrightarrow{L_p} X$ and $Y_n \xrightarrow{L_p} Y$ implies $X_n + Y_n \xrightarrow{L_p} X + Y$, we can show convergence of each of the summand individually. We established $T_n/n \to T^*$ in Corollary 2 and by Proposition 1, we have $\mathbf{E}[C_n]/n \to 19/6$. For the remaining sum, consider the first summand as an example; the others are similar. Using Proposition 1 once more and the independence of V and $(\mathbf{S}, \mathbf{I})$, we find

$$\begin{aligned}
\mathbf{E}\Big[\Big(\frac{1}{n}\mathbf{1}_{\mathcal{E}_1^{(n)}}\mathbf{E}\big[C_{I_1} \mid I_1^{(n)}\big] - \tfrac{19}{6}S_1\mathbf{1}_{\mathcal{E}_1}\Big)^2\Big] &\le \mathbf{E}\Big[\Big(\mathbf{1}_{\mathcal{E}_1^{(n)}}\frac{\frac{19}{6}I_1^{(n)} + o(n)}{n} - \tfrac{19}{6}S_1\mathbf{1}_{\mathcal{E}_1}\Big)^2\Big] \\
&\le \big(\tfrac{19}{6}\big)^2\Big\|\tfrac{I_1^{(n)}}{n}\mathbf{1}_{\mathcal{E}_1^{(n)}} - S_1\mathbf{1}_{\mathcal{E}_1}\Big\|_2 + o(1) \\
&\to 0, \qquad \text{for } n \to \infty,
\end{aligned}$$

where the last step follows from the L_2-convergence shown above for the coefficients.

Cond. (B) We have to show that

$$\mathbf{E}\Big[\sum_{j=1}^{3} |S_j \mathbf{1}_{\mathcal{E}_j}|^2\Big] \;<\; 1\,.$$

By linearity of the expectation, it suffices to consider the summands in isolation. We compute

$$\begin{aligned}\mathbf{E}\big[\big(S_1\mathbf{1}_{\mathcal{E}_1}\big)^2\big] &= \int_{s_1=0}^{1}\int_{s_2=0}^{1-s_1}\int_{v=0}^{1} s_1^2\mathbf{1}_{\{v<s_1\}}\cdot 2\;dv\;ds_2\;ds_1 \\ &= \int_{s_1=0}^{1}(1-s_1)\int_{v=0}^{s_1} 2s_1^2\;dv\;ds_1 \;=\; 2\int_{x=0}^{1}(1-x)x^3\;dx \;=\; \tfrac{1}{10}\,.\end{aligned}$$

The other summands are symmetric, so we find $\mathbf{E}\big[\sum_{j=1}^{3}|S_j\mathbf{1}_{\mathcal{E}_j}|^2\big] = \frac{3}{10} < 1$.

Cond. (C) The third condition of Theorem B.1 requires for $j = 1, 2, 3$ and any constant $c \in \mathbb{N}$ that

$$\mathbf{E}\left[\mathbf{1}_{\{I_j^{(n)}\le c \,\vee\, I_j^{(n)}=n\}}\Big|\tfrac{I_j^{(n)}}{n}\mathbf{1}_{\mathcal{E}_j^{(n)}}\Big|^2\right] \;\to\; 0\,, \qquad \text{as } n\to\infty\,.$$

As we have $I_j^{(n)} \le n-2$ by definition and $\big|\frac{I_j^{(n)}}{n}\mathbf{1}_{\mathcal{E}_j^{(n)}}\big|^2$ is bounded from above by 1 uniformly in n, it suffices to show

$$\mathbf{Prob}\big(I_j^{(n)} \le c\big) \;\to\; 0\,.$$

But this directly follows from the weak law of large numbers as the expected values grow linearly with n, that is, $\mathbf{E}[I_j^{(n)} \,|\, \mathbf{S}] = nS_j$.

All conditions of Theorem B.1 are fulfilled, so we obtain $C_n^\circ \xrightarrow{\zeta_2} C^\circ$ and the distribution of C° is the unique fixed point of the distributional equation

$$C^\circ \;\overset{\mathcal{D}}{=}\; S_1\mathbf{1}_{\mathcal{E}_1}C^\circ \;+\; S_2\mathbf{1}_{\mathcal{E}_2}C^{\circ\prime} \;+\; S_3\mathbf{1}_{\mathcal{E}_3}C^{\circ\prime\prime} \;+\; T^* - \tfrac{19}{6} \;+\; \tfrac{19}{6}\sum_{j=1}^{3} S_j\mathbf{1}_{\mathcal{E}_j} \tag{19}$$

among all centered distributions with finite second moments; $C^{\circ\prime}$ and $C^{\circ\prime\prime}$ are independent copies of C°.

It remains to transfer convergence of C_n° to C_n^*. By Proposition 1, we have

$$C_n^* \;=\; C_n^\circ + \tfrac{19}{6} + o(1)\,.$$

So, the asymptotic difference between the two is a (deterministic) constant and C_n^* converges to the limit $C^* = C^\circ + 19/6$. Inserting this into the limit equation (19) for C°, we obtain (1) and the first part of Theorem 1 is proved.

C. Proof of Convergence: Extreme Ranks

For extreme ranks, we can apply Theorem 5.1 of [30], restated here for convenience:

Theorem C.1 ([30, Theorem 5.1])**.** *Let $(Y_n)_{n\in\mathbb{N}}$ be a sequence of s-integrable, $0 < s \le 3$, random variables satisfying the recurrence*

$$Y_n \;\overset{\mathcal{D}}{=}\; \sum_{r=1}^{K} Y_{I_r^{(n)}}^{(r)} \;+\; b_n\,, \qquad n \ge n_0\,,$$

where $K \in \mathbb{N}$ is a constant, $(b_n, I^{(n)})$ and $(Y_n^{(1)})_{n\in\mathbb{N}}$, …, $(Y_n^{(K)})_{n\in\mathbb{N}}$ are independent and $Y_i^{(r)}$ is distributed like Y_i for all $r = 1,\dots,K$ and $i \ge 0$. Additionally, we require $\mathbf{Prob}[I_r^{(n)} = n] \to 0$ as $n\to\infty$ and $\mathbf{Var}[Y_n] > 0$ for $n \ge n_0$. Assume further that there are functions $f, g : \mathbb{N}_0 \to \mathbb{R}_{\ge 0}$ such that (a) if $0 < s \le 1$ we have $g(n) > 0$ for large n; (b) if $1 < s \le 2$ additionally $\mathbf{E}[Y_n] = f(n) + o\big(\sqrt{g(n)}\big)$ holds and (c) if $2 < s \le 3$ additionally $\mathbf{Var}[Y_n] = g(n) + o\big(g(n)\big)$ is satisfied. Moreover, assume the following conditions:

(A) *For all* $r = 1, \dots, K$ *holds* $\sqrt{g(I_r^{(n)})/g(n)} \xrightarrow{L_s} A_r^*$;

(B) $\frac{1}{\sqrt{g(n)}}\big(b_n - f(n) + \sum_{r=1}^K f(I_r^{(n)})\big) \xrightarrow{L_s} b^*$;

(C) $\mathbf{E}\big[\sum_{r=1}^K (A_r^*)^s\big] < 1$.

Then for (X_n) *defined by* $X_n := \big(Y_n - f(n)\big)/\sqrt{g(n)}$, *we have* $\lim_{n\to\infty} \zeta_s(X_n, X) = 0$ *for a random variable* X *whose distribution is given as the unique fixed point of*

$$X \;\stackrel{\mathcal{D}}{=}\; \sum_{r=1}^K A_r^* X^{(r)} \;+\; b^*$$

among all distributions with (a) finite s*th moments, if* $0 < s \le 1$, *(b) finite* s*th moments and mean zero, if* $1 < s \le 2$, *and (c) finite* s*th moments, variance one and mean zero, if* $2 < s \le 3$. *Here,* $(A_1^*, \dots, A_K^*, b^*)$ *and* $X^{(1)}, \dots, X^{(K)}$ *are independent and* $X^{(r)}$ *is distributed like* X *for* $r = 1, \dots, K$. □

This theorem is a special case of theorem we used in Appendix B and applies to distributional recurrences where the (non-normalized) costs Y_n of subproblems directly contribute, without a factor in front of them. Our equation (15) is of this form.

We apply Theorem C.1 with $K = 1$, $s = 2$ and functions $f(n) = \frac{19}{8}n$ and $g(n) = n^2$. By Proposition 2 we have $\mathbf{E}[\hat{C}_n] = f(n) + o\big(\sqrt{g(n)}\big)$ as needed. Theorem C.1 then states convergence of the centered variables $\hat{C}_n^\circ = \frac{1}{n}(\hat{C}_n - \frac{19}{8}n)$, given conditions (A), (B) and (C) are fulfilled. We check the conditions:

Cond. (A) We have by Lemma 4:

$$\sqrt{\frac{g(I_1^{(n)})}{g(n)}} \;=\; \frac{I_1^{(n)}}{n} \;\xrightarrow{L_2}\; S_1 \,.$$

Cond. (B) Using Proposition 2 and Corollary 2, we compute:

$$g(n)^{-1/2}\big(T_n - f(n) + f(I_1^{(n)})\big) \;=\; T_n^* - \tfrac{19}{8} + \tfrac{19}{8}\frac{I_1^{(n)}}{n} \;\xrightarrow{L_2}\; T^* + \tfrac{19}{8}(S_1 - 1)\,.$$

Cond. (C) A simple computation shows $\mathbf{E}\big[S_1^2\big] = \frac{1}{6} < 1$.

All requirements of Theorem C.1 are fulfilled and we conclude that

$$\hat{C}_n^\circ \;=\; \frac{\hat{C}_n - \frac{19}{8}n}{n} \;=\; \hat{C}_n^* - \tfrac{19}{8} \;\xrightarrow{\zeta_2}\; \hat{C}^\circ\,, \tag{20}$$

where the distribution of $\hat{C}^\circ$ is obtained as unique fixed point of

$$\hat{C}^\circ \;\stackrel{\mathcal{D}}{=}\; S_1 C^\circ \;+\; T^* + \tfrac{19}{8}(S_1 - 1)\,, \tag{21}$$

among all centered distributions with finite second moments. The constant difference between $\hat{C}_n^*$ and $\hat{C}_n^\circ$ implies that $\hat{C}_n^*$ converges to $C^\circ + \frac{19}{8}$. Inserting $C^* = C^\circ + \frac{19}{8}$ in (21) we obtain equation (4); the first part of Theorem 2 is proved.