Probabilistic Analysis of the Dual-Pivot Quicksort "Count"

Ralph Neininger and Jasmin Straub

Institute for Mathematics

Goethe University Frankfurt

Germany

E-Mail: {`neininger, jstraub`}`@math.uni-frankfurt.de`

Abstract

Recently, Aumüller and Dietzfelbinger proposed a version of a dual-pivot quicksort, called "Count", which is optimal among dual-pivot versions with respect to the average number of key comparisons required. In this note we provide further probabilistic analysis of "Count". We derive an exact formula for the average number of swaps needed by "Count" as well as an asymptotic formula for the variance of the number of swaps and a limit law. Also for the number of key comparisons the asymptotic variance and a limit law are identified. We also consider both complexity measures jointly and find their asymptotic correlation.

1 Introduction

In 2009, Oracle replaced the sorting algorithm quicksort in its Java 7 runtime library by a new dual-pivot quicksort due to Vladimir Yaroslavskiy. This was based on the good performance of the new dual-pivot version in experiments. Since then, various theoretical studies were devoted to explain, quantify, generalize and improve the dual-pivot quicksort, starting with a rigorous average case analysis of Wild and Nebel [10] for the numbers of comparisons and swaps in a uniform probabilistic model. However, classical cost parameters turned out to not be sufficient to explain the good performance of the new dual-pivot quicksort. Hence, authors also took the memory hierarchy of computer storage into account and studied cache effects, see Kushagra et al. [5], and quantities related to cache misses, in particular the number of scanned elements, see Nebel et al. [7] and [3].

The basic idea to gain while using a dual-pivot quicksort is that during the partitioning stages elements may not need to be compared to both pivot elements. Thus, the theoretical question on how to optimally arrange the partitioning stages arose. Regarding the average number of key comparisons this has fully been solved in Aumüller et al. [2] where the dual-pivot quicksort version "Count" is identified to be optimal in this respect.

The dual-pivot quicksort "Count" uses two pivot elements which are chosen from the array as the first and the last element. Assume they are p and q with $p < q$. Then all other elements are compared to the pivot elements to classify them as smaller as p, between p and q (called medium) or being larger than q. Obviously, elements between p and q need to be compared to p and q to get classified. However, if an element smaller than p is first compared to p, there is no need for also comparing it to q, the same for elements larger than q if they are first compared to q. Now, to typically save some comparisons one keeps track on how many elements smaller than p and larger than q already have been identified. If you have classified more small than large elements so far, this indicates that the data are split unevenly by p and q with more small elements than large elements. Hence, in such a case, one compares the next element first to p, and vice versa. After partitioning the elements according to this rule the algorithm recurses on the lists of elements smaller than p, medium, and larger than q, see Algorithm 1.

The aim of the present note is to provide further probabilistic analysis of "Count" in the model of uniformly permuted (distinct) data. While there are little mathematical innovations required to study such quicksort variants there are still new combinatorial structures to reveal and to study. Also, we feel that it is worth adding probabilistic analysis for "Count" due to its distinguished role. We study the numbers of comparisons and swaps required by "Count" with respect to expectations, variances, limit laws and correlations. Along our analysis also the number of scanned elements could be studied. In Section 2 our results are stated. Section 3 contains the analysis and proofs of the theorems. At the end the version of "Count" used in the present note is stated explicitly (Algorithm 1) which differs slightly from the original version of Aumüller et al. [2].

The results of this note are part of the second mentioned author's master's thesis [9].

2 Results

Throughout, we assume that the data are distinct and uniformly permuted.

2.1 Mean values The mean value for the number $\mathcal{C}_n$ of key comparisons has already been derived in Aumüller et al. [2, Theorem 12.1]. For $n \geq 4$, we have

$$
\begin{aligned}
\mathbb{E}[\mathcal{C}_n] &= \frac{9}{5}n\mathcal{H}_n - \frac{1}{5}n\mathcal{H}_n^{\mathrm{alt}} - \frac{89}{25}n + \frac{67}{40}\mathcal{H}_n - \frac{3}{40}\mathcal{H}_n^{\mathrm{alt}} \\
&\quad - \frac{83}{800} + \frac{(-1)^n}{10} - \frac{\mathbb{1}_{\{n\ \mathrm{even}\}}}{320}\left(\frac{1}{n-3} + \frac{3}{n-1}\right) \\
&\quad + \frac{\mathbb{1}_{\{n\ \mathrm{odd}\}}}{320}\left(\frac{3}{n-2} + \frac{1}{n}\right) \\
&= \frac{9}{5}n\log(n) + A_c n + \frac{67}{40}\log(n) + \mathrm{O}(1),
\end{aligned}
$$

as $n \to \infty$, where $A_c = \frac{9}{5}\gamma + \frac{1}{5}\log(2) - \frac{89}{25} = -2.382...$,

$$
\mathcal{H}_n := \sum_{k=1}^n \frac{1}{k} \qquad \text{and} \qquad \mathcal{H}_n^{\mathrm{alt}} := \sum_{k=1}^n \frac{(-1)^k}{k}.
$$

Note, that, as $n \to \infty$, we have

$$
\mathcal{H}_n = \log(n) + \gamma + \mathrm{O}\left(\frac{1}{n}\right), \quad \mathcal{H}_n^{\mathrm{alt}} = -\log(2) + \mathrm{O}\left(\frac{1}{n}\right),
$$

and $\gamma = 0.577...$ denotes the Euler–Mascheroni constant.

We add the corresponding formula for the mean number of swaps.

THEOREM 2.1. *For the number of swaps $\mathcal{S}_n$ of the dual-pivot quicksort "Count" in Algorithm 1 when sorting a random permutation of length n, for $n \geq 4$, we have*

$$
\begin{aligned}
\mathbb{E}[\mathcal{S}_n] &= \frac{3}{4}n\mathcal{H}_n + \frac{1}{20}n\mathcal{H}_n^{alt} - \frac{4}{5}n + \frac{3}{4}\mathcal{H}_n + \frac{1}{20}\mathcal{H}_n^{alt} \\
&\quad - \frac{23}{160} - \frac{(-1)^n}{40} - \frac{\mathbb{1}_{\{n\ \mathrm{even}\}}}{320}\left(\frac{1}{n-3} + \frac{3}{n-1}\right) \\
&\quad + \frac{\mathbb{1}_{\{n\ \mathrm{odd}\}}}{320}\left(\frac{3}{n-2} + \frac{1}{n}\right) \\
&= \frac{3}{4}n\log(n) + A_s n + \frac{3}{4}\log(n) + \mathrm{O}(1),
\end{aligned}
$$

as $n \to \infty$, with $A_s = \frac{3}{4}\gamma - \frac{1}{20}\log(2) - \frac{4}{5} = -0.401....$

The derivation of the formula for $\mathbb{E}[\mathcal{S}_n]$ in Theorem 2.1 uses a surprising combinatorial identity for which we have not yet found an intuitive explanation, see Proposition 3.1.

2.2 Deviations and correlations For the orders of the standard deviations and correlations we have the following asymptotic results which are independent of the special implementation, i.e., valid for our version Algorithm 1 as well as for the version in Aumüller et al. [2].

THEOREM 2.2. *For the numbers of key comparisons $\mathcal{C}_n$ and swaps $\mathcal{S}_n$ of the dual-pivot quicksort "Count" when sorting a random permutation of length n, as $n \to \infty$, we have*

$$
\begin{aligned}
\mathrm{Var}(\mathcal{C}_n) &\sim \sigma_c^2 n^2, \\
\mathrm{Var}(\mathcal{S}_n) &\sim \sigma_s^2 n^2, \\
\mathrm{Cov}(\mathcal{C}_n, \mathcal{S}_n) &\sim \sigma_{c,s}^2 n^2,
\end{aligned}
$$

where

$$
\begin{aligned}
\sigma_c^2 &= \frac{1609}{300} - \frac{27}{50}\pi^2 + \frac{3}{10}\log 2 = 0.241\dots, \\
\sigma_s^2 &= \frac{47}{48} - \frac{3}{32}\pi^2 + \frac{3}{32}\log 2 = 0.118\dots, \\
\sigma_{c,s}^2 &= \frac{43}{20} - \frac{9}{40}\pi^2 + \frac{7}{40}\log 2 = 0.050\dots.
\end{aligned}
$$

Hence, the asymptotic correlation between $\mathcal{C}_n$ and $\mathcal{S}_n$ is

$$
\mathrm{Corr}(\mathcal{C}_n, \mathcal{S}_n) \sim 0.298755\dots \quad (n \to \infty).
$$

Note that for the classic quicksort (with one pivot element and the partitioning method of Hoare and Sedgewick, see, e.g., Wild [13, Algorithm 2]) we have a strong negative correlation of about -0.864 between the numbers of key comparisons and key exchanges, see [8]. The different nature of correlations for the classic quicksort and the dual-pivot quicksort "Count" is already transparent in the scatter plot in Figure 1.
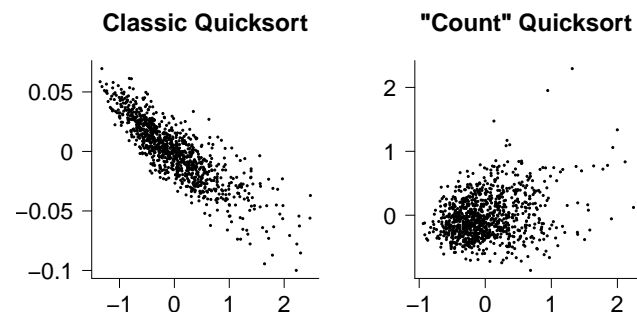

Figure 1: Scatter plots of the normalized versions of $\mathcal{C}_n$ and $\mathcal{S}_n$ as in Theorem 2.3 for the classic quicksort (left) and the dual-pivot quicksort "Count" (right). Depicted are 1000 samples of random permutations of size $n = 10000$.

2.3 Limit distributions The asymptotic behaviour of the distributions of $\mathcal{C}_n$ and $\mathcal{S}_n$ is determined by the recursive distributional equation (RDE) for probability distributions on $\mathbb{R}^2$,

$$(2.1) \qquad X \stackrel{d}{=} \sum_{r=1}^{3} \begin{pmatrix} D_r & 0 \\ 0 & D_r \end{pmatrix} X^{(r)} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

with

$$b_1 = 1 + D_2 + \min\{D_1, D_3\} + \frac{9}{5} \sum_{r=1}^{3} D_r \log D_r,$$

$$b_2 = D_1 + D_3 + \mathbb{1}_{\{D_3 > D_1\}} \Big(\frac{1}{2} D_1 + D_2 - D_3\Big)$$

$$\qquad + \frac{3}{4} \sum_{r=1}^{3} D_r \log D_r,$$

where $X^{(1)}$, $X^{(2)}$, $X^{(3)}$ and (D_1, D_2, D_3) are independent, $X^{(r)}$ is distributed as X for $r \in \{1, 2, 3\}$ and (D_1, D_2, D_3) is distributed as the spacings

$$(2.2) \qquad (U_1 \wedge U_2, U_1 \vee U_2 - U_1 \wedge U_2, 1 - U_1 \vee U_2)$$

of two independent, unif$[0,1]$ distributed random variables U_1 and U_2 in $[0,1]$, where $\wedge$ and $\vee$ denote min and max, respectively. Among all centered distributions with a finite second moment RDE (2.1) has a unique solution, see [8, Lemma 3.1]. We denote this solution by

$$(2.3) \qquad \mathcal{L}(X) = \mathcal{L}\begin{pmatrix} X_c \\ X_s \end{pmatrix}.$$

Note that the first and second coordinate of RDE (2.1) give univariate RDEs which also characterize the marginals $\mathcal{L}(X_c)$ and $\mathcal{L}(X_s)$, respectively.

THEOREM 2.3. *For the normalized numbers of key comparisons and swaps of the dual-pivot quicksort "Count" when sorting a random permutation of length n we have jointly in distribution, as $n \to \infty$,*

$$\left(\frac{\mathcal{C}_n - \mathbb{E}[\mathcal{C}_n]}{n}, \frac{\mathcal{S}_n - \mathbb{E}[\mathcal{S}_n]}{n}\right) \stackrel{d}{\longrightarrow} (X_c, X_s),$$

where (X_c, X_s) is given in (2.3).

From the form of (2.1) we obtain for the limits X_c and X_s the existence of densities.

COROLLARY 2.1. *The distributions of X_c and X_s have densities each being infinitely smooth and rapidly decreasing (together with all their higher derivatives).*

3 Analysis

In this section, we sketch the analysis leading to our results. Recall that we assume that the array is uniformly distributed. For simplicity, we assume that the input consists of a sequence $(U_i)_{i \geq 1}$ of independent, uniformly on $[0,1]$ distributed random variables and that, for fixed n, we have to sort the array $[U_1, \ldots, U_n]$. To further simplify the analysis we instead of choosing U_1 and U_n as pivot elements, cf. Algorithm 1, choose U_1 and U_2 as pivot elements. (This does not change the distributions of our quantities but implies that various convergences also hold in a strong (almost sure, L_2) sense.)

We start with the growth of the numbers of elements smaller than p, between p and q, and larger than q during the partitioning stage. We denote by (S_i, M_i, L_i) the vector of these numbers after i elements have been compared to the pivot elements, $i \in \{0, \ldots, n-2\}$. We first analyse an urn model which captures the growth dynamics of $(S_i, M_i, L_i)_{i \in \{0, \ldots, n-2\}}$.

A related urn model. Consider a Pólya–Eggenberger urn model with three types $\mathfrak{s}$, $\mathfrak{m}$ and $\mathfrak{l}$ of balls (corresponding to "small", "medium" and "large" elements). Initially, the urn contains one ball of each type. In each step a ball is drawn uniformly from the urn (and independently from the earlier draws) and returned to the urn together with one ball of the same type, i.e., the replacement matrix is the 3×3 identity matrix. Denote by (S_i', M_i', L_i') the vector of numbers of balls of each type added during steps $j \in \{1, \ldots, i\}$. In other words, the composition of the urn after i steps consists of $S_i' + 1$ elements of type $\mathfrak{s}$, of $M_i' + 1$ elements of type $\mathfrak{m}$, and of $L_i' + 1$ elements of type $\mathfrak{l}$. It is easy to see and has been used earlier, see [2] and [13], that for each $n \geq 3$ the processes $(S_i', M_i', L_i')_{i \in \{0, \ldots, n-2\}}$ and $(S_i, M_i, L_i)_{i \in \{0, \ldots, n-2\}}$ are equal in distribution. Since we are only analysing parameters which are identified by distributions we hence drop the primes and shortly write (S_i, M_i, L_i) for (S_i', M_i', L_i'). We need the following conditional probability of adding a ball of type $\mathfrak{l}$ while there are more balls of type $\mathfrak{l}$ than of type $\mathfrak{s}$.

PROPOSITION 3.1. *For all $i \geq 1$ we have*

$$\mathbb{P}(L_{i+1} = L_i + 1 \mid L_i > S_i) = \frac{1}{2}.$$

Proof. It is easy to see that for each $i \geq 0$ the vector (S_i, M_i, L_i) is uniformly distributed over its possible values, i.e., for all $s, m, \ell \geq 0$ with $s + m + \ell = i$, we have

$$(3.4) \qquad \mathbb{P}((S_i, M_i, L_i) = (s, m, \ell)) = \frac{2}{(i+1)(i+2)}.$$

This implies that $\mathbb{P}(L_i = S_i)$ is $1/(i+1)$ if i is even and $1/(i+2)$ if i is odd. Hence, by symmetry, we obtain

$$(3.5) \qquad \mathbb{P}(L_i > S_i) = \begin{cases} \frac{i}{2(i+1)}, & \text{if } i \text{ is even,} \\ \frac{i+1}{2(i+2)}, & \text{if } i \text{ is odd.} \end{cases}$$

On the other hand, by the law of total probability and (3.4), we have

$$\mathbb{P}(L_{i+1} = L_i + 1, L_i > S_i)$$
$$= \sum_{\ell=1}^{i} \sum_{\substack{s=0 \\ s \leq \ell-1}}^{i-\ell} \Big(\mathbb{P}(S_i = s, M_i = i - s - \ell, L_i = \ell)$$
$$\times \mathbb{P}(L_{i+1} = L_i + 1 \mid S_i = s, M_i = i - s - \ell, L_i = \ell) \Big)$$
$$= \sum_{\ell=1}^{i} \sum_{\substack{s=0 \\ s \leq \ell-1}}^{i-\ell} \frac{2}{(i+1)(i+2)} \cdot \frac{\ell+1}{i+3}$$
$$= \begin{cases} \frac{i}{4(i+1)}, & \text{if } i \text{ is even,} \\ \frac{i+1}{4(i+2)}, & \text{if } i \text{ is odd.} \end{cases}$$

Combining (3.5) and the latter identity implies the assertion.

Expected values. To study the expectation of the number of swaps $\mathcal{S}_n$ denote, for $n \geq 2$, by S_n^+, M_n^+ and L_n^+ the number of small, medium and large elements compared to q first when partitioning an input sequence of length n with the dual-pivot quicksort "Count". Formally, we have

$$S_n^+ = \sum_{i=1}^{n-3} \mathbb{1}_{\{L_i > S_i, \, S_{i+1} = S_i + 1\}},$$
$$M_n^+ = \sum_{i=1}^{n-3} \mathbb{1}_{\{L_i > S_i, \, M_{i+1} = M_i + 1\}},$$
$$L_n^+ = \sum_{i=1}^{n-3} \mathbb{1}_{\{L_i > S_i, \, L_{i+1} = L_i + 1\}},$$

since $S_0 = L_0 = 0$, which means that the first element is always compared to p first. Furthermore, we denote by $I^{(n)} = (I_1^{(n)}, I_2^{(n)}, I_3^{(n)})$ the sizes of the three sublists generated in the first partitioning stage. Note, that we have $\frac{1}{n} I^{(n)} \to (D_1, D_2, D_3)$ in L_p for any $1 \leq p < \infty$ as $n \to \infty$ (the U_1 and U_2 in (2.2) are now the pivot elements).

The number $T_S(n)$ of swaps executed by the dual-pivot "Count" during the first partitioning stage is composed by

▷ $I_1^{(n)} + \frac{1}{2} S_n^+$ swaps for the small elements (since there is a swap for each small element compared to p first

and a *rotate3*-operation, i.e., $3/2$ swaps, for each small element compared to q first),

▷ $I_3^{(n)} - L_n^+$ swaps for the large elements compared to p first (note that the large elements which are compared to q first do not need to be swapped),

▷ M_n^+ swaps for the medium elements compared to q first (the medium elements which are compared to p first do not need to be swapped), and

▷ two swaps at the end in order to bring the pivots to their final positions (both, line 37 and line 39 of Algorithm 1 need two write-accesses to the array).

Hence, we have

$$(3.6) \quad T_S(n) = 2 + I_1^{(n)} + I_3^{(n)} + \frac{1}{2} S_n^+ + M_n^+ - L_n^+.$$

We first derive the expectation of $T_S(n)$, see (3.8).

LEMMA 3.1. *For all $n \geq 2$ we have*

$$\mathbb{E}[S_n^+] = \mathbb{E}[L_n^+ - M_n^+] = \frac{1}{12} n - \frac{7}{24} + \frac{1}{8(n - \mathbb{1}_{\{n \text{ even}\}})}.$$

Proof. Similarly to the proof of Proposition (3.1) we find

$$\mathbb{P}(L_i > S_i, S_{i+1} = S_i + 1)$$
$$= \sum_{\ell=1}^{i} \sum_{\substack{s=0 \\ s \leq \ell-1}}^{i-\ell} \frac{2}{(i+1)(i+2)} \cdot \frac{s+1}{i+3}$$
$$(3.7) \qquad = \begin{cases} \frac{i(i+4)}{12(i+1)(i+3)}, & \text{if } i \text{ is even,} \\ \frac{1}{12}, & \text{if } i \text{ is odd.} \end{cases}$$

By definition,

$$\mathbb{E}[S_n^+] = \mathbb{E}\left[\sum_{i=1}^{n-3} \mathbb{1}_{\{L_i > S_i, \, S_{i+1} = S_i + 1\}} \right]$$
$$= \sum_{i=1}^{n-3} \mathbb{P}(L_i > S_i, \, S_{i+1} = S_i + 1).$$

Now, we plug in expression (3.7) on the right hand side of the latter display and use induction. This implies the claimed expression for $\mathbb{E}[S_n^+]$. Furthermore, using

4

Proposition 3.1 we have

$$
\mathbb{E}[S_n^+ + M_n^+ - L_n^+]
$$

$$
= \sum_{i=1}^{n-3} \Big(\mathbb{P}(L_i > S_i,\ S_{i+1} = S_i + 1)
$$

$$
+ \ \mathbb{P}(L_i > S_i,\ M_{i+1} = M_i + 1)
$$

$$
- \ \mathbb{P}(L_i > S_i,\ L_{i+1} = L_i + 1) \Big)
$$

$$
= \sum_{i=1}^{n-3} \mathbb{P}(L_i > S_i) \Big(\mathbb{P}(L_{i+1} = L_i \,|\, L_i > S_i)
$$

$$
- \ \mathbb{P}(L_{i+1} = L_i + 1 \,|\, L_i > S_i) \Big)
$$

$$
= 0,
$$

hence $\mathbb{E}[S_n^+] = \mathbb{E}[L_n^+ - M_n^+]$.

Since by symmetry we have $\mathbb{E}[I_r^{(n)}] = (n-2)/3$ for $r \in \{1, 2, 3\}$ we obtain from (3.6) and Lemma 3.1 that

$$
(3.8) \qquad \mathbb{E}[T_S(n)] = \frac{5}{8}n + \frac{13}{16} - \frac{1}{16(n - \mathbb{1}_{\{n \text{ even}\}})}.
$$

We are now prepared to prove Theorem 2.1.

Proof. (Theorem 2.1) We draw back to earlier analysis of dual-pivot quantities which led to similar recurrences for their expectations with other toll functions, see Wild [12, Section 4.2.1]. For $n \geq 2$, we have the recurrence

$$
\mathbb{E}[\mathcal{S}_n] = \frac{6}{n(n-1)} \sum_{k=0}^{n-2} (n-k-1)\mathbb{E}[\mathcal{S}_k] + \mathbb{E}[T_S(n)],
$$

with $\mathbb{E}[\mathcal{S}_0] = \mathbb{E}[\mathcal{S}_1] = 0$. From Wild's work we know that, for $n \geq 4$,

$$
\mathbb{E}[\mathcal{S}_n] = \frac{1}{\binom{n}{4}} \sum_{i=5}^{n} \binom{i}{4} \sum_{j=3}^{i-2} \Big(\mathbb{E}[T_S(j+2)]
$$

$$
- \frac{2j}{j+2}\,\mathbb{E}[T_S(j+1)] + \frac{\binom{j}{2}}{\binom{j+2}{2}}\mathbb{E}[T_S(j)] \Big)
$$

$$
(3.9) \qquad + \frac{n+1}{5} \left(\mathbb{E}[T_S(4)] + \frac{1}{2}\mathbb{E}[T_S(2)] \right).
$$

Now, the affine component $\frac{5}{8}n + \frac{13}{16}$ of $\mathbb{E}[T_S(n)]$ in (3.8) implies an overall contribution to $\mathbb{E}[\mathcal{S}_n]$, cf. [12, Section 4.2.1.1], of

$$
(3.10) \qquad \frac{3}{4}n\mathcal{H}_n + \frac{3}{4}\mathcal{H}_n - \frac{33}{40}n - \frac{27}{160}.
$$

To identify the contribution of summand $(16(n - \mathbb{1}_{\{n \text{ even}\}}))^{-1}$ of $\mathbb{E}[T_S(n)]$ in (3.8) to $\mathbb{E}[\mathcal{S}_n]$ we set $\mu_n :=$

$(n - \mathbb{1}_{\{n \text{ even}\}})^{-1}$. In view of (3.9) we need to compute

$$
(3.11)
$$

$$
\frac{1}{\binom{n}{4}} \sum_{i=5}^{n} \binom{i}{4} \sum_{j=3}^{i-2} \Big(\mu_{j+2} - \frac{2j}{j+2}\mu_{j+1} + \frac{\binom{j}{2}}{\binom{j+2}{2}}\mu_j \Big).
$$

For the inner sum in (3.11) we find, by plugging in the values of μ_j, μ_{j+1} and μ_{j+2} and using induction, for $i \geq 5$, that

$$
\sum_{j=3}^{i-2} \left(\mu_{j+2} - \frac{2j}{j+2}\mu_{j+1} + \frac{\binom{j}{2}}{\binom{j+2}{2}}\mu_j \right)
$$

$$
= -4\mathcal{H}_i^{\text{alt}} + (-1)^i \frac{2}{i} - \frac{17}{6}.
$$

Plugging this expression into (3.11) and another induction imply, for $n \geq 4$, that

$$
\frac{1}{\binom{n}{4}} \sum_{i=5}^{n} \binom{i}{4} \left(-4\mathcal{H}_i^{\text{alt}} + \frac{2(-1)^i}{i} - \frac{17}{6} \right)
$$

$$
= -\frac{17}{30}n - \frac{4}{5}n\mathcal{H}_n^{\text{alt}} - \frac{4}{5}\mathcal{H}_n^{\text{alt}} - \frac{17}{30} + \frac{2}{5}(-1)^n
$$

$$
+ \frac{\mathbb{1}_{\{n \text{ even}\}}}{20} \left(\frac{1}{n-3} + \frac{3}{n-1} \right)
$$

$$
- \frac{\mathbb{1}_{\{n \text{ odd}\}}}{20} \left(\frac{3}{n-2} + \frac{1}{n} \right).
$$

Hence, the total contribution of summand $(16(n - \mathbb{1}_{\{n \text{ even}\}}))^{-1}$ of $\mathbb{E}[T_S(n)]$ to $\mathbb{E}[\mathcal{S}_n]$ is

$$
\frac{1}{16} \Bigg(-\frac{2}{5}n - \frac{4}{5}(n+1)\mathcal{H}_n^{\text{alt}} - \frac{2}{5} + \frac{2}{5}(-1)^n
$$

$$
+ \frac{\mathbb{1}_{\{n \text{ even}\}}}{20} \left(\frac{1}{n-3} + \frac{3}{n-1} \right)
$$

$$
- \frac{\mathbb{1}_{\{n \text{ odd}\}}}{20} \left(\frac{3}{n-2} + \frac{1}{n} \right) \Bigg).
$$

Together with (3.10) this implies the assertion.

An alternative route to the formula for $\mathbb{E}[\mathcal{S}_n]$ in Theorem 2.1 is via generating functions as used in [2] to derive $\mathbb{E}[\mathcal{C}_n]$, see [9] for details.

Variances, correlation and limit laws. The results stated in Theorems 2.2 and 2.3 follow from a standard application of the contraction method based on the expansions of $\mathbb{E}[\mathcal{C}_n]$ and $\mathbb{E}[\mathcal{S}_n]$ mentioned at the beginning of Section 2.1 and in Theorem 2.1, respectively. We refer the reader to Section 4 of [11] where the application of the contraction method to dual-pivot quicksort is explained and worked out for parameters with other toll

functions. To apply this approach to $\mathcal{C}_n$ and $\mathcal{S}_n$ we only need to derive the asymptotic L_2-behaviour of our toll functions in the recurrences for $\mathcal{C}_n$ and $\mathcal{S}_n$.

Note, that the number of key comparisons executed during the first partitioning stage of the dual-pivot quicksort "Count" is given by

$$T_C(n) = n - 1 + I_2^{(n)} + I_3^{(n)} + S_n^+ - L_n^+.$$

For the normalized quantities $X_0 := (0,0)^t$ and

$$(3.12) \qquad X_n := \left(\frac{\mathcal{C}_n - \mathbb{E}[\mathcal{C}_n]}{n}, \frac{\mathcal{S}_n - \mathbb{E}[\mathcal{S}_n]}{n} \right)^t$$

for $n \geq 1$, we have the distributional recurrence

$$(3.13) \qquad X_n \overset{d}{=} \sum_{r=1}^{3} A_r^{(n)} X_{I_r^{(n)}}^{(r)} + b^{(n)}$$

for $n \geq 2$, where $(X_j^{(1)})_{j \in \{0,\dots,n\}}$, $(X_j^{(2)})_{j \in \{0,\dots,n\}}$, $(X_j^{(3)})_{j \in \{0,\dots,n\}}$ and $(b^{(n)}, I^{(n)})$ are independent, $X_j^{(r)}$ is distributed as X_j for $r \in \{1,2,3\}$, $j \geq 0$ and

$$A_r^{(n)} = \frac{1}{n} \begin{pmatrix} I_r^{(n)} & 0 \\ 0 & I_r^{(n)} \end{pmatrix},$$

$$b^{(n)} = \frac{1}{n} \begin{pmatrix} T_C(n) - \mathbb{E}[\mathcal{C}_n] + \sum_{r=1}^{3} \mathbb{E}[\mathcal{C}_{I_r^{(n)}} | I_r^{(n)}] \\ T_S(n) - \mathbb{E}[\mathcal{S}_n] + \sum_{r=1}^{3} \mathbb{E}[\mathcal{S}_{I_r^{(n)}} | I_r^{(n)}] \end{pmatrix}.$$

Recall that we have the L_2-convergence $A_r^{(n)} \to \mathrm{diag}(D_r, D_r)$ for $r \in \{1,2,3\}$. A standard calculation based on the expansions of $\mathbb{E}[\mathcal{C}_n]$ and $\mathbb{E}[\mathcal{S}_n]$ implies that

$$\frac{1}{n} \left(-\mathbb{E}[\mathcal{C}_n] + \sum_{r=1}^{3} \mathbb{E}[\mathcal{C}_{I_r^{(n)}} | I_r^{(n)}] \right) \xrightarrow{L_2} \frac{9}{5} \sum_{r=1}^{3} D_r \log D_r,$$

$$\frac{1}{n} \left(-\mathbb{E}[\mathcal{S}_n] + \sum_{r=1}^{3} \mathbb{E}[\mathcal{S}_{I_r^{(n)}} | I_r^{(n)}] \right) \xrightarrow{L_2} \frac{3}{4} \sum_{r=1}^{3} D_r \log D_r.$$

Hence, it remains to identify the limits of $T_C(n)/n$ and $T_S(n)/n$ which reduces to the limit of $\frac{1}{n}(S_n^+, M_n^+, L_n^+)$.

LEMMA 3.2. *As $n \to \infty$, we have*

$$\frac{1}{n} S_n^+ \xrightarrow{L_2} \mathbb{1}_{\{D_3 > D_1\}} D_1,$$

$$\frac{1}{n} M_n^+ \xrightarrow{L_2} \mathbb{1}_{\{D_3 > D_1\}} D_2,$$

$$\frac{1}{n} L_n^+ \xrightarrow{L_2} \mathbb{1}_{\{D_3 > D_1\}} D_3.$$

Proof. We consider the random lattice path $(W_i)_{i \geq 0}$ defined by $W_0 := 0$ and, for $i \geq 1$,

$$W_i = \sum_{j=1}^{i} \left(\mathbb{1}_{\{\text{the } j\text{-th classified element is large}\}} - \mathbb{1}_{\{\text{the } j\text{-th classified element is small}\}} \right),$$

i.e., W_i is the difference of the number of large and small elements after having classified i elements. Conditionally on $(D_1, D_2, D_3) = (d_1, d_2, d_3)$, the process $(W_i)_{i \geq 0}$ is a simple random walk on $\mathbb{Z}$ going one step down with probability d_1, staying at its current state with probability d_2 and going one step up with probability d_3.

We first consider the case $d_1 < d_3$. Here, the random walk $(W_i)_{i \geq 0}$ has a positive drift. From the strong law of large numbers, we obtain that W_i tends to $+\infty$ almost surely. This implies that on $\{D_3 > D_1\}$, there exists almost surely some random $n_0 \in \mathbb{N}$ such that $W_i > 0$ for all $i \geq n_0$. This means that from index n_0 on we always compare to pivot q first. Hence, on $\{D_3 > D_1\}$ we obtain

$$\frac{I_1^{(n)} - n_0}{n} \leq \frac{S_n^+}{n} \leq \frac{I_1^{(n)}}{n}.$$

Thus, on $\{D_3 > D_1\}$, we have that S_n^+/n converges to D_1 almost surely.

Similarly, if $d_1 > d_3$ the random walk $(W_i)_{i \geq 0}$ has a negative drift and we find $S_n^+/n \to 0$ almost surely on $\{D_3 < D_1\}$.

Overall, using dominated convergence, we find

$$\frac{S_n^+}{n} \xrightarrow{L_2} \mathbb{1}_{\{D_3 > D_1\}} D_1.$$

The convergences of the other two components in the formulation of the present lemma follow similarly.

Based on the L_2-convergences derived in the present subsection the results of Theorems 2.2 and 2.3 follow along the lines of Section 4 of [11].

Corollary 2.1 follows from a general theorem of Leckey [6] based on techniques of Fill and Janson [4].

References

Algorithm 1 Dual-Pivot Quicksort Algorithm "Count". Slight modifications to the version of Aumüller et al. [2, Algorithm 1] are marked and commented on in blue color.

```
 1: procedure COUNT(A, left, right)
 2:     if right ≤ left then
 3:         return
 4:     if A[right] < A[left] then
 5:         p ← A[right]; q ← A[left]
 6:         // in [2]: swap A[left] and A[right]
 7:     else
 8:         p ← A[left]; q ← A[right]
 9:         // in [2]: p ← A[left]; q ← A[right] instead of ll. 7/8
10:     i ← left + 1; k ← right − 1; j ← i
11:     d ← 0
12:     while j ≤ k do
13:         if d ≥ 0 then
14:             if A[j] < p then
15:                 swap A[i] and A[j]
16:                 i ← i + 1; j ← j + 1; d ← d + 1
17:             else
18:                 if A[j] < q then
19:                     j ← j + 1
20:                 else
21:                     swap A[j] and A[k]
22:                     k ← k − 1; d ← d − 1
23:         else
24:             if A[k] > q then
25:                 k ← k − 1; d ← d − 1
26:             else
27:                 if A[k] < p then
28:                     // Perform a cyclic rotation to
29:                     // the left, i.e.,
30:                     // tmp ← A[k]; A[k] ← A[j];
31:                     // A[j] ← A[i]; A[i] ← tmp
32:                     rotate3(A[k], A[j], A[i])
33:                     i ← i + 1; d ← d + 1
34:                 else
35:                     swap A[j] and A[k]
36:                 j ← j + 1
37:     A[left] ← A[i − 1] and A[i − 1] ← p
38:     // in [2]: swap A[left] and A[i − 1]
39:     A[right] ← A[k + 1] and A[k + 1] ← q
40:     // in [2]: swap A[right] and A[k + 1]
41:     COUNT(A, left, i − 2)
42:     COUNT(A, i, k)
43:     COUNT(A, k + 2, right)
```

[1] M. Aumüller and M. Dietzfelbinger (2015) Optimal Partitioning for Dual-Pivot Quicksort, *ACM Trans. Algorithms* **12**, Art. 18, 36 pp.

[2] M. Aumüller, M. Dietzfelbinger, C. Heuberger, D. Krenn, and H. Prodinger (2016+) *Dual-Pivot Quicksort: Optimality, Analysis and Zeros of Associated Lattice Paths.* Preprint. URL http://arxiv.org/abs/1611.00258.

[3] M. Aumüller, M. Dietzfelbinger, and P. Klaue (2016) How Good Is Multi-Pivot Quicksort? *ACM Trans. Algorithms* **13**, Art. 8, 47 pp.

[4] J. A. Fill and S. Janson (2000) Smoothness and decay properties of the limiting Quicksort density function. *Mathematics and Computer Science (Versailles, 2000)*, 53–64. Trends Math., Birkhäuser, Basel.

[5] S. Kushagra, A. López-Ortiz, A. Qiao, and J. I. Munro (2014) Multi-pivot quicksort: Theory and experiments. In: McGeoch, C.C., Meyer, U. (ed.) SIAM Meeting on Algorithm Engineering and Experiments (ALENEX), pp. 47–60.

[6] K. Leckey (2016+) On Densities for Solutions to Stochastic Fixed Point Equations. Preprint. URL http://arxiv.org/abs/1604.05787.

[7] M. E. Nebel, S. Wild, and C. Martínez (2016) Analysis of pivot sampling in dual-pivot Quicksort: a holistic analysis of Yaroslavskiy's partitioning scheme. *Algorithmica*, **75**, 632–683.

[8] R. Neininger (2001) On a Multivariate Contraction Method for Random Recursive Structures with Applications to Quicksort. *Random Structures Algorithms* **19**, 498–524.

[9] J. Straub (2017) *Probabilistic Analysis of Dual-Pivot Quicksort "Count".* Master's Thesis. Goethe University, Frankfurt a.M. URN urn:nbn:de:hebis:30:3-444927

[10] S. Wild and M. E. Nebel (2012) Average Case Analysis of Java 7's Dual Pivot Quicksort. In Leah Epstein and Paolo Ferragina, editors, *European Symposium on Algorithms 2012*, volume 7501 of LNCS, pp. 825–836. Springer.

[11] S. Wild, M. E. Nebel, and R. Neininger (2015) Average Case and Distributional Analysis of Dual-Pivot Quicksort, *ACM Trans. Algorithms* **11**, Art. 22, 42 pp.

[12] S. Wild (2012) *Java 7's Dual Pivot Quicksort.* Master's Thesis. University of Kaiserslautern. URL https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/3463

[13] S. Wild (2016) *Dual-Pivot Quicksort and Beyond: Analysis of Multiway Partitioning and Its Practical Potential.* Ph.D. Thesis. University of Kaiserslautern. URL https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/4468