An Extended Note on the Comparison-optimal Dual-Pivot Quickselect

Daniel Krenn*

Abstract

In this note the precise minimum number of key comparisons any dual-pivot quickselect algorithm (without sampling) needs on average is determined. The result is in the form of exact as well as asymptotic formulæ of this number of a comparison-optimal algorithm. It turns out that the main terms of these asymptotic expansions coincide with the main terms of the corresponding analysis of the classical quickselect, but still—as this was shown for Yaroslavskiy quickselect—more comparisons are needed in the dual-pivot variant. The results are obtained by solving a second order differential equation for the generating function obtained from a recursive approach.

1 Introduction

Quickselect [10] (also called "Hoare's find algorithm" or "Hoare's selection algorithm") is an algorithm to select the jth smallest element (the "jth rank") of an unordered list. It uses the same partitioning strategy as quicksort [9, 11, 13]: One element of the list is chosen as a pivot element and the remaining are split into two sublists containing the elements smaller and larger than the pivot. Both algorithms then proceed recursively on the sublists (quicksort) or on one sublist (quickselect).

1.1 Quicksort The classical quicksort algorithm with one pivot element needs $2n \log n + O(n)$, as $n \to \infty$, key comparisons on average to sort a list of length n. Using more than one pivot element can decrease this number. For example, Yaroslavskiy's [24] partitioning strategy and dual-pivot quicksort algorithm results in only $1.9n \log n + O(n)$, see Wild and Nebel [21]. This can be improved further. The lower bound for dual-pivot quicksort is $1.8n \log n + O(n)$ key comparisons; this was shown in Aumüller and Dietzfelbinger [1]. Their optimal/minimal strategy called "Clairvoyant" uses an oracle, and therefore it is non-algorithmic. Its algorithmic version "Count" still only needs $1.8n \log n + O(n)$ key comparisons. The precise analysis of [2] reveals the linear terms of these two strategies, and it is claimed that "Count" is the optimal partitioning strategy.

Note that all strategies considered in this article choose the pivots without sampling. A survey on quicksort with a special focus on dual-pivot partitioning can be found in Wild [20].

1.2 Single-Pivot vs. Dual-Pivot Quickselect We use $H_n = \sum_{k=1}^{n} 1/k$ to denote the harmonic numbers.

Due to the improvements of quicksort with dual-pivoting which were mentioned above, one would expect that a dual-pivot quickselect needs as well fewer key comparisons than the classical quickselect. However, Wild, Nebel and Mahmoud [22] show that this is not true. While the classical quickselect needs

$$\begin{aligned}(1.1)\quad & 3n - 8H_n + 13 - 8n^{-1}H_n \\ &= 3n - 8\log n - 8\gamma + 13 + O\bigl(n^{-1}\log n\bigr)\end{aligned}$$

key comparisons on average when selecting a rank chosen uniformly at random, see Mahmoud, Modarres and Smythe [16], quickselect with Yaroslavskiy's partitioning strategy [22] needs

$$\begin{aligned}(1.2)\quad & \tfrac{19}{6}n - \tfrac{37}{5}H_n + \tfrac{1183}{100} - \tfrac{37}{5}n^{-1}H_n - \tfrac{71}{300}n^{-1} \\ &= \tfrac{19}{6}n - \tfrac{37}{5}\log n - \tfrac{37}{5}\gamma + \tfrac{1183}{100} + O\bigl(n^{-1}\log n\bigr)\end{aligned}$$

key comparisons. The same is true for the average number of key comparisons when selecting the smallest or largest rank. There it increases from

$$(1.3)\qquad 2n - 2H_n = 2n - 2\log n - 2\gamma + O\bigl(n^{-1}\bigr)$$

of the classical quickselect [16] to

$$\begin{aligned}(1.4)\quad & \tfrac{57n^4-48n^3H_n-178n^3+144n^2H_n+135n^2-96nH_n-14n+24}{24n(n-1)(n-2)} \\ &= \tfrac{19}{8}n - 2\log n - 2\gamma - \tfrac{7}{24} + O\bigl(n^{-1}\bigr)\end{aligned}$$

*Email: math@danielkrenn.at *or* daniel.krenn@aau.at, Institut für Mathematik, Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65–67, 9020 Klagenfurt am Wörthersee, Austria. The author is supported by the Austrian Science Fund (FWF): P 24644-N26. The author kindly thanks Helmut Prodinger for his inspiring talk "Quickselect, multiple Quickselect, Quicksort with median-of-three partition and related material" given at AAU Klagenfurt in May 2016.

of Yaroslavskiy's quickselect [22]. The latter reference, as well as [20], provide further discussions and insights.

The question that is answered in this note is: Does any dual-pivot quickselect with the comparison-optimal partitioning strategy beat (in terms of the number of key comparisons) the classical quickselect or not?

1.3 Discussion: The New Results Face to Face with the Existing Results The aim of this note is to determine a lower bound for all dual-pivot quickselect algorithms by counting the number of key comparisons in quickselect using the optimal paritioning strategy "Count".

On the one hand, we analyze selecting a random rank ("grand averages"). This results in

$$(1.5)\quad \overline{C}_n^{\min} = 3n + \frac{3}{20}(\log n)^2 + \left(\frac{\gamma + \log 2}{10} + \frac{319}{50}\right)\log n + O(1)$$

key comparisons on average (expected value), formulated precisely as Theorem 4.1 and Corollary 4.1. As expected, this number of key comparisons is (asymptotically) lower than the number in Yaroslavskiy quickselect (1.2) which has main term $\frac{19}{6}n$. We even get the same main term $3n$ as in the classical quickselect (1.1). Unfortunately the second order term in (1.5) is still larger than the second order term in (1.1). Thus, we can answer the question posed above, whether a dual-pivot quickselect beats the classical quickselect, by "no"—at least when selecting a random rank.

On the other hand, we analyze selecting the jth smallest/largest rank with $j \in \{1, 2, 3, 4\}$ which results in

$$(1.6)\quad C_{n,j}^{\min} = C_{n,n-j+1}^{\min} = \frac{9}{4}n + \frac{1}{12}(\log n)^2 + \left(\frac{\gamma + \log 2}{6} + t_j\right)\log n + O(1)$$

key comparisons on average. There the t_j are explicitly known constants. See Section 5 for details. Again the main term is lower than that of the Yaroslavski variant (1.4), but it is still larger than the main term of the classical quickselect (1.3). So again our main question is answered by a "no".

We also analyze the theoretical (non-algorithmic) "Clairvoyant" partitioning strategy, see [1, 2] and Section 2. It turns out that the main term of the average number of key comparisons is the same as in (1.5) and (2.7) respectively, but surprisingly its second order term has the opposite sign. Thus it needs fewer key comparisons than the classical quickselect (formulæ (1.1) and (1.3)). Details are to be found at the end of Sections 4 and 5.

1.4 What Else? Many other properties and variants of the (classical) quickselect are studied and can be extended to dual-pivot quickselect algorithms and can be investigated for them. Prodinger [18], Lent and Mahmoud [15], Panholzer and Prodinger [17], and Kuba [14] analyze quickselect when selecting multiple ranks simultaneously. Different strategies to choose the pivot are possible as well. For example, Kirschenhofer, Prodinger and Martinez [12] use a median of three strategy.

Distributional results and higher moments such as the variance are also feasible. For Yaroslavskiy's quicksort, this was done by Wild, Nebel and Neininger [23] and for the corresponding quickselect by Wild, Nebel and Mahmoud [22]. It is possible to extend the methods of the latter for our optimal paritioning strategy; this is a task for the full version of this extended abstract.

1.5 Notation: Harmonic Numbers and More Here a short note on the notation used in the sections below. There are

- the harmonic numbers $H_n = \sum_{k=1}^{n} 1/k$ and
- the alternating harmonic numbers $H_n^{\mathrm{alt}} = \sum_{k=1}^{n} (-1)^k/k$.

Moreover, we use

- the Iversonian notation

$$[\mathit{expr}] = \begin{cases} 1 & \text{if } \mathit{expr} \text{ is true,} \\ 0 & \text{if } \mathit{expr} \text{ is false,} \end{cases}$$

which was popularized by Graham, Knuth, and Patashnik [5].

By $\gamma = 0.5772156649\ldots$, we denote the Euler–Mascheroni constant.

2 Partitioning Strategies

As mentioned in the introduction, the average number of comparisons for a dual-pivot quicksort or quickselect algorithm depends on its partitioning strategy. So let us suppose we have an (unsorted) list of distinct elements. We choose the first and the last element as pivot elements p and q. We assume $p < q$; this needs one comparison.

Informally, a partitioning strategy is an algorithm, which, in each step,

1. takes an unclassified element,
2. compares it with p or q first,

3. if not already classified compares it with the remaining element p or q, and

4. marks the element as small ($< p$), medium (between p and q) or large ($> q$).

The choice whether to choose p or q for the first comparison in each step may depend on the history of the outcome of the previous classifications. Additionally the index of the element to read may depend on this history as well. However, the index of the element to read does not have any influence on the results presented in this article.

A more formal definition of partitioning strategies can be found in Aumüller and Dietzfelbinger [1]; they use the following decision trees to model a partitioning strategy: A strategy is described by a complete rooted ternary tree with $n-2$ levels (as $n-2$ elements have to be classified). Each vertex is labeled by a pair consisting of the index of the element to be classified and of p or q indicating which element to use for the first comparison for the classification. The three outgoing edges of a vertex are labeled by small, medium and large, respectively, and represent the outcome of the classification. Every order/permutation of a list of elements corresponds to a path in this tree which starts at the root and ends in a leaf.

Next, we describe a couple of partitioning strategies.

"Smaller pivot first" We always compare with the smaller pivot first. Each small element needs only one comparison to be classified, each medium and each large element needs two comparisons. This results in

$$P_n^{\mathrm{sf}} = \frac{5}{3}n - \frac{7}{3}$$

for the expected number of key comparisions to classify a list of $n \geq 2$ elements. (Two of these list-elements will be the pivots.) The corresponding generating function of the expected cost of partitioning is

$$P^{\mathrm{sf}}(z) = \frac{5}{3(1-z)^2} - \frac{4}{1-z} - \frac{2}{3}(1-z) + 3.$$

See also Appendix D for details. Note that the very same result holds for the "larger pivot first" partitioning strategy by symmetry.

"Yaroslavskiy" ([24]) See the introduction for details and references.

"Count" We keep track of the numbers of already classified small and large elements. If there were more larger than smaller elements up to now, then we use q for the first comparison in the next step, otherwise p.

This is the optimal—meaning that it minimizes the expected number of key comparisons—algorithmic dual-pivot partitioning strategy, see [2]. The expected number of key comparisons to classify a list of n elements (two of these elements will be the pivots) is

$$P_n^{\mathrm{ct}} = \frac{3}{2}n + \frac{1}{4}\log n + \frac{2\gamma + 2\log 2 - 19}{8} + O\big(n^{-1}\big).$$

It was analyzed in [2], where an exact formula and a precise asymptotic expansion was stated. The corresponding generating function of the expected cost of partitioning is known explicitly as

$$P^{\mathrm{ct}}(z) = \frac{3}{2(1-z)^2} + \frac{\operatorname{artanh}(z)}{2(1-z)} - \frac{31z^2}{8(1-z)} - \frac{3+z}{8}\operatorname{artanh}(z) - \frac{3}{2} - \frac{25z}{8} \tag{2.7}$$

from [2] as well.

This article's main focus is on the partitioning strategy "Count".

"Clairvoyant" This strategy uses an oracle to predict the number of small and large elements in the remaining (unsorted) list. If there are going to be more larger than smaller elements, then we use q for the first comparison, otherwise p.

Note that this strategy is not algorithmic. It provides a theoretic lower bound for the number of key comparisons of all partitioning strategies [1]. Again, an explicit analysis can be found in [1] and [2]. The expected number of key comparisons to classify a list of n elements (two of these elements will be the pivots) is

$$P_n^{\mathrm{cv}} = \frac{3}{2}n - \frac{1}{4}\log n - \frac{2\gamma + 2\log 2 + 13}{8} + O\big(n^{-1}\big).$$

When using these strategies for quickselect, randomness in the obtained sublists after the partitioning step is preserved. We refer here to Wild, Nebel and Mahmoud [22], who use a criterion of Hennequin [7]. See also the third volume of the book of Knuth [13].

3 The Recurrence

Let $n \in \mathbb{N}_0$. We assume that the input of our quickselect algorithm is a random permutation of $\{1, \dots, n\}$ chosen uniformly at random. For $j \in \{1, \dots, n\}$, let us denote by $C_{n,j}$ the average number of comparisons needed to select the jth smallest element.

By symmetry of the algorithm, selecting the jth largest element costs as much as selecting the jth smallest element, thus we have

$$C_{n,j} = C_{n,n-j+1}. \tag{3.8}$$

The average number of comparisons satisfies the following recurrence.

Proposition 3.1. *Let $j \in \{1, \dots, n\}$. Then*

$$C_{n,j} = P_n + S_{n,j} + M_{n,j} + L_{n,j}$$

with

$$S_{n,j} = \frac{1}{\binom{n}{2}} \sum_{s=j}^{n-2} (n-1-s) C_{s,j},$$

$$M_{n,j} = \frac{1}{\binom{n}{2}} \sum_{m=1}^{n-2} \sum_{s=\max\{0,j-m-1\}}^{\min\{j-2,n-m-2\}} C_{m,j-s-1},$$

$$L_{n,j} = \frac{1}{\binom{n}{2}} \sum_{\ell=n-j+1}^{n-2} (n-1-\ell) C_{\ell,n-j+1},$$

for $n \geq 2$, and $C_{0,j} = 0$ and $C_{1,j} = 0$.

The special case of the recurrence for $j = 1$ can be found in [22]. There, a recurrence for analyzing the grand averages is presented as well.

Proof of Proposition 3.1. We assume that the input is a random permutation of $\{1, \dots, n\}$. The expected cost $C_{n,j}$ is the sum of the expected partitioning cost P_n and the sum of the cost of the recursive call for the small elements $S_{n,j}$, medium elements $M_{n,j}$ or large elements $L_{n,j}$. Throughout this proof, the random variables of the number of small, medium and large elements are denoted by S, M and L, respectively, and we have $n - 2 = S + M + L$.

After the partitioning step, we proceed with the small elements if the number S of small elements is at least j; this number can be at most $n - 2$ because of the two pivots p and q. For a fixed realization S, there are $n - 1 - S$ possibilities—all of them are equally likely—to partition the medium and large elements. This results in the probability $\mathbb{P}(S = s) = (n-1-s)/\binom{n}{2}$ to continue with selecting the jth smallest element of a list of s elements; the expected cost for this is $C_{s,j}$. The quantity $S_{n,j}$ follows by summing up over all s.

Similarly, the number L of large elements has to be at least $n - j + 1$ to recurs into the large-branch. There are $n-1-L$ possibilities, thus $\mathbb{P}(L = \ell) = (n-1-\ell)/\binom{n}{2}$ for every ℓ. For a fixed ℓ, we need to find the $(j-n+\ell)$th smallest element (as $n - \ell = s + m + 2$), so the cost is $C_{\ell,j-n+\ell} = C_{\ell,n-j+1}$ by symmetry (3.8). The result for $L_{n,j}$ follows.

In order to recurs on the medium elements, we need S to be at most $j-2$ and L to be at most $n-j-1$; both have 0 as a lower bound. All events are equally likely which results in the probability $\mathbb{P}(S = s, L = \ell) = 1/\binom{n}{2}$. The expected cost is $C_{n-2-s-\ell,j-s-1}$ as $m = n-2-\ell$ and we continue to find the $(j-s-1)$st element. Summing up and rewriting the resulting double sum in terms of the indices s and m (instead of s and ℓ) yields $M_{n,j}$. This completes the proof.

We translate the recurrence above into the world of generating functions. We set $C(z,u) = \sum_{n,j} C_{n,j} z^n u^j$, and, for the number of comparisons for partitioning, we define $P(z) = \sum_n P_n z^n$.

The symmetry (3.8) translates to the functional equation

$$\begin{aligned} u\,C(zu, 1/u) &= \sum_{n,j} C_{n,j} z^n u^{n-j+1} \\ &= \sum_{n,j} C_{n,n-j+1} z^n u^j = C(z,u)\,. \end{aligned} \tag{3.9}$$

We need this functional equation in the proof below. The generating function obtained by the recurrence of Proposition 3.1 satisfies the following ordinary differential equation in the variable z.

Proposition 3.2. *We have*

$$\frac{\partial^2}{\mathrm{d}z^2} C(z,u) = \frac{u}{1-u} \left(P''(z) - u^2 P''(zu)\right) + 2\,C(z,u)\,r(z,u)$$

with

$$r(z,u) = \frac{1}{(1-z)^2} + \frac{u}{(1-z)(1-zu)} + \frac{u^2}{(1-zu)^2}.$$

If $u = 1$, then we have

$$\frac{\partial^2}{\mathrm{d}z^2} C(z,u)\Big|_{u=1} = \frac{1}{z}\left(z^2 P''(z)\right)' + \frac{6}{(1-z)^2} C(z,1)\,.$$

Note that a generating function and an ordinary differential equation for the grand averages—this is equivalent to considering $C(z,1)$—for the particular Yaroslavskiy quickselect can be found in [22].

The full proof of Proposition 3.2 can be found in Appendix A.

Sketch of the proof of Proposition 3.2. We use the recurrence of Proposition 3.1 to obtain

$$
\begin{aligned}
n(n-1)C_{n,j} &= n(n-1)P_n[1 \le j \le n] \\
&+ 2\sum_{s=0}^{n-1}(n-1-s)C_{s,j} + 2\sum_{m=0}^{n-2}\sum_{s=0}^{n-m-2} C_{m,j-s-1} \\
&+ 2\sum_{\ell=0}^{n-1}(n-1-\ell)C_{\ell,n-j+1}.
\end{aligned}
$$

We multiply by $z^{n-2}u^j$ and sum up over all $n \ge 2$ and all j; we treat each summand separately, so we have an equation $\mathcal{C} = \mathcal{P} + \mathcal{S} + \mathcal{M} + \mathcal{L}$.

The parts $\mathcal{C}$ and $\mathcal{P}$ are straight forward to determine.

Next, we deal with $\mathcal{S}$. We extend the sum by including $n = 1$, then shift from $n-1$ to n, and get

$$
\begin{aligned}
\mathcal{S} &= 2\sum_j \sum_{n\ge 2}\sum_{s=0}^{n-1}(n-1-s)C_{s,j}z^{n-2}u^j \\
&= 2\sum_j \sum_{n\ge 1}\sum_{s=0}^{n-1}(n-1-s)C_{s,j}z^{n-2}u^j \\
&= 2\sum_j \sum_{n\ge 0}\sum_{s=0}^{n}(n-s)z^{n-s-1}C_{s,j}z^s u^j.
\end{aligned}
$$

Rewriting the convolution to a product of generating functions yields

$$
\begin{aligned}
\mathcal{S} &= 2\Big(\sum_{n\ge 0} n z^{n-1}\Big)\sum_j\sum_{n\ge 0} C_{n,j}z^n u^j \\
&= 2\Big(\frac{1}{1-z}\Big)' C(z,u) = \frac{2}{(1-z)^2}\, C(z,u)\,.
\end{aligned}
$$

We proceed in a similar manner with $\mathcal{L}$, where (3.9) has to be used. To deal with the sum $\mathcal{M}$, we have to take into account one additional summation; we succeed by proceeding as above. The overall result follows as $\mathcal{C} = \mathcal{P} + \mathcal{S} + \mathcal{M} + \mathcal{L}$.

4 A Random Selection

We focus on the partitioning strategy "Count", see Section 2 for details, which minimizes the number of key comparisons among all dual-pivot partitioning strategies.

Let $n \in \mathbb{N}_0$ be fixed. In this section, we assume that j is an integer of $\{1,\dots,n\}$ chosen uniformly at random. This means for our algorithm, that we perform a random selection. The input is again a random permutation of $\{1,\dots,n\}$. We study the expected value/average number $\overline{C}_n^{\mathrm{ct}}$ of key comparisons of this selection depending on the input size n; the following theorem holds.

Theorem 4.1. *The average number (expected value) of key comparisons in the comparison-optimal dual-pivot quickselect algorithm—it uses strategy "Count"—when performing a random selection is*

$$
\begin{aligned}
\overline{C}_n^{\mathrm{ct}} &= 3n + \frac{3}{20n}\sum_{k=1}^{n-1} H_k H_{n-k} \\
&\quad - \frac{3}{10n}\sum_{k=1}^{n}\frac{H_{k-1}^{\mathrm{alt}}}{k}(n-k+1) \\
&\quad - \frac{194}{25}H_n + \frac{9}{25}H_n^{\mathrm{alt}} + \frac{1564}{125} \\
&\quad - \frac{1527}{200}\frac{H_n}{n} + \frac{47}{200}\frac{H_n^{\mathrm{alt}}}{n} + \frac{783}{4000n} - \frac{9}{50}\frac{(-1)^n}{n} \\
&\quad + \frac{22}{1600n}\Bigg(\frac{n-1}{n(n-2)}[n \text{ odd}] \\
&\qquad\qquad - \frac{n-5}{(n-1)(n-3)}[n \text{ even}]\Bigg)
\end{aligned}
$$

for $n \ge 4$.

We have $\overline{C}_0 = \overline{C}_1 = 0$, $\overline{C}_2 = 8/3$ and $\overline{C}_3 = 9/2$. We extract the asymptotic behavior out of the generating function used in the proof of Theorem 4.1; this is the corollary below.

Corollary 4.1. *The average number (expected value) of key comparisons in the comparison-optimal dual-pivot quickselect algorithm—it uses strategy "Count"—when performing a random selection is*

$$
\overline{C}_n^{\mathrm{ct}} = 3n + \frac{3}{20}(\log n)^2 + \left(\frac{\gamma + \log 2}{10} + \frac{319}{50}\right)\log n + O(1)
$$

asymptotically as n tends to infinity.

Proof of Theorem 4.1 and Corollary 4.1. Proposition 3.2 provides an ordinary differential equation for $C(z,1)$. As this linear differential equation is basically the same—it only differs in the inhomogeneity—as for the dual-pivot quicksort, its solution is

$$
\begin{aligned}
&(4.10)\quad C(z,1) = \\
&\quad (1-z)^3\int_0^z (1-t)^{-6}\int_0^t (1-s)^3\frac{1}{s}\big(s^2 P''(s)\big)'\,ds\,dt
\end{aligned}
$$

as described in Wild [19] (who follows Hennequin [8]; see also [2] for the explicit solution).

We use $P(z) = P^{\mathrm{ct}}(z)$ (and write $C^{\mathrm{ct}}(z,1)$ instead of $C(z,1)$). By performing the integration (4.10), we obtain the generating function

$$
\begin{aligned}
C^{\mathrm{ct}}(z,1) &= \frac{6}{(1-z)^3} + \frac{3\,\log(1-z)^2}{20\,(1-z)^2} \\
&\quad - \frac{3}{10(1-z)^2}\,L_2(z) + \frac{194\,\log(1-z)}{25\,(1-z)^2} \\
&\quad - \frac{9\,\log(1+z)}{25\,(1-z)^2} - \frac{531}{125\,(1-z)^2} + \frac{\log(1+z)}{8\,(1-z)} \\
&\quad - \frac{\log(1-z)}{8\,(1-z)} - \frac{1389}{800\,(1-z)} \\
&\quad - \frac{11}{3200}\,(1-z)^3\log(1-z) \\
&\quad + \frac{11}{3200}\,(1-z)^3\log(1+z) \\
&\quad - \frac{29}{750}\,(1-z)^3 + \frac{11}{1600}\,(1-z)^2 \\
&\quad - \frac{11}{1600}\,z + \frac{77}{4800}.
\end{aligned}
$$

Here we use the abbreviation

$$
L_2(z) = -\int_0^z \frac{\log(1+t)}{1-t}\mathrm{d}t,
$$

see Appendix B. Theorem 4.1 follows by extracting the coefficients of the generating function exactly, whereas Corollary 4.1 follows by extracting the coefficients asymptotically via singularity analysis [3, 4]. Appendix B might assist.

The authors of [1] and [2] study the partitioning strategy "Clairvoyant" which is based on an oracle, see Section 2 for details. Our methods here can be easily modified to obtain results for this strategy as well.

Theorem 4.2. *The average number (expected value) of key comparisons in the dual-pivot quickselect algorithm with strategy "Clairvoyant" when performing a random selection is*

$$
\begin{aligned}
\overline{C}_n^{\mathrm{cv}} &= 3n - \frac{3}{20}\sum_{k=1}^{n-1} H_k H_{n-k} \\
&\quad + \frac{3}{10}\frac{1}{n}\sum_{k=1}^{n}\frac{H_{k-1}^{\mathrm{alt}}}{k}(n-k+1) \\
&\quad - \frac{196}{25}H_n - \frac{9}{25}H_n^{\mathrm{alt}} + \frac{1576}{125} \\
&\quad - \frac{1593}{200}\frac{H_n}{n} - \frac{47}{200}\frac{H_n^{\mathrm{alt}}}{n} - \frac{703}{4000}\frac{1}{n} + \frac{9}{50}\frac{(-1)^n}{n} \\
&\quad + \frac{22}{1600}\frac{1}{n}\biggl(\frac{n-1}{n(n-2)}[n \text{ odd}] \\
&\qquad\qquad - \frac{n-5}{(n-1)(n-3)}[n \text{ even}]\biggr).
\end{aligned}
$$

This equals

$$
\begin{aligned}
\overline{C}_n^{\mathrm{cv}} &= 3n - \frac{3}{20}(\log n)^2 \\
&\qquad + \left(-\frac{3\gamma + 3\log 2}{10} + \frac{461}{50}\right)\log n + O(1)
\end{aligned}
$$

asymptotically as n tends to infinity.

The proof of Theorem 4.2 can be found in Appendix C.

For completeness, we include the expected value/average number of key comparisons for dual-pivot quickselect with the partitioning strategy "smaller pivot first" here. Note that these results are equal to those of the strategy "larger pivot first" by symmetry.

Proposition 4.1. *The average number (expected value) of key comparisons in the dual-pivot quickselect algorithm with strategy "smaller pivot first" when performing a random selection is*

$$
\overline{C}_n^{\mathrm{sf}} = \frac{10}{3}n - \frac{44}{5}H_n + \frac{354}{25} - \frac{44}{5}\frac{H_n}{n} + \frac{2}{75}.
$$

This equals

$$
\overline{C}_n^{\mathrm{sf}} = \frac{10}{3}n + \frac{44}{5}\log n + \frac{44}{5}\gamma - \frac{758}{75} + \frac{12}{5}n^{-1} + O\bigl(n^{-2}\bigr)
$$

asymptotically as n tends to infinity.

5 Selecting the jth Smallest/Largest Element

In this section, we determine the expected value/average number of key comparisons for selecting, among others, the smallest ($j = 1$) or largest element ($j = n$) of a random permutation of $\{1, \ldots, n\}$, all equally likely. Again we use the partitioning strategy "Count" (Section 2).

We use the bivariate generating function $C(z,u)$ of Section 3. Let $j \in \{1, \ldots, n\}$, and let us group $C(z,u)$ in terms of the parameter j as

$$
C(z,u) = \sum_{j\ge 1} C_j(z)\,u^j.
$$

We extract the jth coefficient of the differential equation for $C(z,u)$ of Proposition 3.2. This leads to the following system of ordinary differential equations. Note that $C_1(z)$ in the case of Yaroslavskiy quickselect is stated in [22].

Lemma 5.1. *We have*

$$
C_j''(z) - \frac{2}{(1-z)^2}\,C_j(z) = Q_j(z)
$$

with

$$Q_j(z) = P''(z) - \sum_{n<j} n(n-1)P_n z^{n-2} + 2\sum_{k=0}^{j-1} C_k(z)\, z^{j-k-2}\left(\frac{z}{1-z} + j - k - 1\right)$$

and $C_j(0) = C_j'(0) = 0$.

The proof is straight forward and can be found in Appendix C.

Remark 5.1. *The ordinary differential equation*

$$C''(z) - \frac{2}{(1-z)^2}\, C(z) = Q(z)$$

with $C(0) = C'(0) = 0$ *has the solution*

$$C(z) = (1-z)^2 \int_0^z (1-t)^{-4} \int_0^t (1-s)^2\, Q(s)\, \mathrm{d}s\mathrm{d}t. \tag{5.11}$$

This provides a way to solve for $C_j(z)$ *of Lemma 5.1.*

The proof of Remark 5.1 can be found in Appendix C.

We are now able to obtain cost coefficients as stated in the following proposition.

Proposition 5.1. *The average number (expected value) of key comparisons in the comparison-optimal dual-pivot quickselect algorithm—it uses strategy "Count"—when selecting the smallest or largest element is*

$$C^{\mathrm{ct}}_{n,1} = C^{\mathrm{ct}}_{n,n} = \frac{9}{4}n + \frac{1}{12}\sum_{k=1}^{n-1}\frac{H_k}{n-k} - \frac{1}{6}\sum_{k=2}^{n}\frac{H^{\mathrm{alt}}_{k-1}}{k} - \frac{43}{18}H_n + \frac{1}{18}H^{\mathrm{alt}}_n + \frac{5}{108} + \frac{[n\ \mathrm{odd}]\,(n-1)}{36n(n-2)} - \frac{[n\ \mathrm{even}]}{36(n-1)}.$$

Note that one can rewrite this exact formula, in particular $\sum_{k=1}^{n-1} H_k/(n-k)$, in terms of other variants of the harmonic numbers, see [6] or the original work of Zave [25].

Corollary 5.1. *The average number (expected value) of key comparisons in the comparison-optimal dual-pivot quickselect algorithm—it uses strategy "Count"—when selecting the smallest or largest element is*

$$C^{\mathrm{ct}}_{n,1} = C^{\mathrm{ct}}_{n,n} = \frac{9}{4}n + \frac{1}{12}(\log n)^2 + \left(\frac{\gamma + \log 2}{6} + \frac{7}{3}\right)\log n + O(1)$$

asymptotically as n *tends to infinity.*

Proof of Proposition 5.1 and Corollary 5.1. Again we use $P(z) = P^{\mathrm{ct}}(z)$ and write $C^{\mathrm{ct}}_j(z)$ instead of $C_j(z)$. Solving the differential equation of Lemma 5.1 by Remark 5.1 results in the generating function

$$C^{\mathrm{ct}}_1(z) = \frac{9}{4}\frac{1}{(1-z)^2} + \frac{1}{12}\frac{(\log(1-z))^2}{1-z} - \frac{1}{6}\frac{L_2(z)}{1-z} + \frac{7}{3}\frac{\log(1-z)}{1-z} - \frac{1}{18}\frac{1}{1-z}\log\left(\frac{1+z}{1-z}\right) - \frac{119}{54}\frac{1}{1-z} + \frac{1}{72} + \frac{1}{72}(1-z) + \frac{1}{144}(1-z)^2\log\left(\frac{1+z}{1-z}\right) - \frac{2}{27}(1-z)^2.$$

To finish the proofs, we extract the coefficients, see also Appendix B.

The system of ordinary differential equations of Lemma 5.1 can be solved iteratively. We calculate the coefficients $C^{\mathrm{ct}}_{n,j}$ and $C^{\mathrm{ct}}_{n,n-j+1}$ with $j \in \{2,3,4\}$ asymptotically in the following proposition. Exact formulæ and the proofs can be found in Appendix C.

Note that it is possible to extend the result to $j = O(1)$ by collecting terms in each iteration; again a task for the full version of this extended abstract.

Proposition 5.2. *The average number (expected value) of key comparisons in the comparison-optimal dual-pivot quickselect algorithm—it uses strategy "Count"—when selecting the first* ($j = 1$)*, second* ($j = 2$)*, third* ($j = 3$) *and fourth* ($j = 4$) *smallest or largest element is*

$$C^{\mathrm{ct}}_{n,j} = C^{\mathrm{ct}}_{n,n-j+1} = \frac{9}{4}n + \frac{1}{12}(\log n)^2 + \left(\frac{\gamma + \log 2}{6} + t_j\right)\log n + O(1)$$

asymptotically as n *tends to infinity with*

$t_1 = \frac{7}{3} = 2.333\ldots$, $\quad t_2 = 1$,
$t_3 = -\frac{3}{10} = -0.3$, $\quad t_4 = -\frac{29}{8} = -3.625$.

Note that Proposition 5.2 superseds Corollary 5.1. The proof of Proposition 5.2 can be found in Appendix C.

As in the section above, we state the corresponding formulæ for the "Clairvoyant" partitioning strategy as well.

Proposition 5.3. *The average number (expected value) of key comparisons in the dual-pivot quickselect algorithm with strategy "Clairvoyant" when selecting the smallest or largest element is*

$$C^{\mathrm{cv}}_{n,1} = C^{\mathrm{cv}}_{n,n} = \frac{9}{4}n - \frac{1}{12}\sum_{k=1}^{n-1}\frac{H_k}{n-k} + \frac{1}{6}\sum_{k=2}^{n}\frac{H^{\mathrm{alt}}_{k-1}}{k} - \frac{41}{18}H_n - \frac{1}{18}H^{\mathrm{alt}}_n + \frac{1}{108} - \frac{1}{72}\frac{[n\ \mathrm{odd}]}{n-2} + \frac{1}{36}\frac{[n\ \mathrm{even}]}{n-1} - \frac{1}{72}\frac{[n\ \mathrm{odd}]}{n}$$

This equals

$$C^{\mathrm{cv}}_{n,1} = C^{\mathrm{cv}}_{n,n} = \frac{9}{4}n - \frac{1}{12}(\log n)^2 + \left(-\frac{\gamma + \log 2}{6} + \frac{7}{3}\right)\log n + O(1)$$

asymptotically as n tends to infinity.

Again, the proof of Proposition 5.3 can be found in Appendix C.

And, again, as in the section above, we state the corresponding formulæ for the "smaller pivot first" partitioning strategy as well; details of the proof can be found in Appendix D.

PROPOSITION 5.4. *The average number (expected value) of key comparisons in the dual-pivot quickselect algorithm with strategy "smaller pivot first" when selecting the smallest or largest element is*

$$C^{\mathrm{sf}}_{n,1} = C^{\mathrm{sf}}_{n,n} = \frac{5}{2}n - \frac{8}{3}H_n + \frac{1}{18}.$$

This equals

$$C^{\mathrm{sf}}_{n,1} = C^{\mathrm{sf}}_{n,n} = \frac{5}{2}n + \frac{8}{3}\log n + \frac{8}{3}\gamma - \frac{22}{9} - \frac{4}{3}n^{-1} + O\big(n^{-2}\big)$$

asymptotically as n tends to infinity.

Appendix

The appendices can be found at `arXiv:1607.05008v2`.

References

[1] Martin Aumüller and Martin Dietzfelbinger, *Optimal partitioning for dual-pivot quicksort*, ACM Trans. Algorithms **12** (2015), no. 2, 18:1–18:36.

[2] Martin Aumüller, Martin Dietzfelbinger, Clemens Heuberger, Daniel Krenn, and Helmut Prodinger, *Counting zeros in random walks on the integers and analysis of optimal dual-pivot quicksort*, Proceedings of the 27th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, 2016, arXiv:1602.04031 [math.CO].

[3] Philippe Flajolet and Andrew Odlyzko, *Singularity analysis of generating functions*, SIAM J. Discrete Math. **3** (1990), 216–240.

[4] Philippe Flajolet and Robert Sedgewick, *Analytic combinatorics*, Cambridge University Press, Cambridge, 2009.

[5] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete mathematics. A foundation for computer science*, second ed., Addison-Wesley, 1994.

[6] Daniel H. Greene and Donald E. Knuth, *Mathematics for the analysis of algorithms*, third ed., Progress in Computer Science and Applied Logic, vol. 1, Birkhäuser Boston, Inc., Boston, MA, 1990.

[7] Pascal Hennequin, *Combinatorial analysis of quicksort algorithm*, RAIRO Inform. Théor. Appl. **23** (1989), no. 3, 317–333.

[8] Pascal Hennequin, *Analyse en moyenne d'algorithmes: tri rapide et arbres de recherche*, Ph.D. thesis, Ecole Polytechnique, Palaiseau, 1991.

[9] Charles A. R. Hoare, *Algorithm 64: Quicksort*, Commun. ACM **4** (1961), no. 7, 321.

[10] ———, *Algorithm 65: find*, Commun. ACM **4** (1961), no. 7, 321–322.

[11] ———, *Quicksort*, Comput. J. **5** (1962), no. 1, 10–15.

[12] Peter Kirschenhofer, Helmut Prodinger, and Conrado Martínez, *Analysis of Hoare's FIND algorithm with median-of-three partition*, Random Structures Algorithms **10** (1997), no. 1-2, 143–156, Average-case analysis of algorithms (Dagstuhl, 1995).

[13] Donald E. Knuth, *The art of computer programming. Vol. 3: Sorting and searching*, second ed., Addison-Wesley, Reading, MA, 1998.

[14] Markus Kuba, *On quickselect, partial sorting and multiple quickselect*, Information Processing Letters **99** (2006), no. 5, 181–186.

[15] Janice Lent and Hosam M. Mahmoud, *Average-case analysis of multiple Quickselect: an algorithm for finding order statistics*, Statist. Probab. Lett. **28** (1996), no. 4, 299–310.

[16] Hosam M. Mahmoud, Reza Modarres, and Robert T. Smythe, *Analysis of QUICKSELECT: an algorithm for order statistics*, RAIRO Inform. Théor. Appl. **29** (1995), no. 4, 255–276.

[17] Alois Panholzer and Helmut Prodinger, *A generating functions approach for the analysis of grand averages for multiple QUICKSELECT*, Proceedings of the Eighth International Conference "Random Structures and Algorithms" (Poznan, 1997), vol. 13, 1998, pp. 189–209.

[18] Helmut Prodinger, *Multiple Quickselect—Hoare's Find algorithm for several elements*, Inform. Process. Lett. **56** (1995), no. 3, 123–129.

[19] Sebastian Wild, *Java 7's dual pivot quicksort*, Master's thesis, University of Kaiserslautern, 2013, `https://kluedo.ub.uni-kl.de/files/3463/wild-master-thesis.pdf`, p. 171.

[20] ———, *Dual-pivot quicksort and beyond: Analysis of multiway partitioning and its practical potential*, Ph.D. thesis, University of Kaiserslautern, 2016, p. 367.

[21] Sebastian Wild and Markus E. Nebel, *Average case analysis of Java 7's dual pivot Quicksort*, Algorithms—ESA 2012, Lecture Notes in Comput. Sci., vol. 7501, Springer, Heidelberg, 2012, pp. 825–836.

[22] Sebastian Wild, Markus E. Nebel, and Hosam Mahmoud, *Analysis of quickselect under Yaroslavskiy's dual-pivoting algorithm*, Algorithmica **74** (2016), no. 1, 485–506.

[23] Sebastian Wild, Markus E. Nebel, and Ralph Neininger, *Average case and distributional analysis of dual-pivot quicksort*, ACM Transactions on Algorithms **11** (2015), no. 3, 22.

[24] Vladimir Yaroslavskiy, *Replacement of quicksort in java.util.arrays with new dual-pivot quicksort*, `http://permalink.gmane.org/gmane.comp.java.openjdk.core-libs.devel/2628`, 2009, Archived version of the discussion in the OpenJDK mailing list.

[25] Derek A. Zave, *A series expansion involving the harmonic numbers*, Information Processing Lett. **5** (1976), no. 3, 75–77.