# Moves and displacements of particular elements in Quicksort ⋆

Conrado Martínez [*,1]

*Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain*

Helmut Prodinger [2]

*Department of Mathematics, University of Stellenbosch, 7602 Stellenbosch, South Africa*

**Abstract**

In this research note we investigate the number of moves and the displacement of particular elements along the execution of the well-known quicksort algorithm. This type of analysis is useful if the costs of a data move were dependent on the source and target locations, and possibly the moved element itself.

From the mathematical point of view, the analysis of these quantities turns out to be related to the analysis of quickselect, a selection algorithm which is a variant of quicksort that finds the $i$-th smallest element of $n$ given elements, without sorting them. Our results constitute thus a novel application of M. Kuba's machinery (*Infor. Proc. Lett.* 99(5):181–186, 2006) for the solution of general quickselect recurrences.

## 1 Introduction

The main goal of this short research note is to present a detailed analysis of the moves of particular elements along the execution of the well-known *quicksort*

⋆ This research was done during mutual visits of the authors to Univ. Stellenbosch and Univ. Pol. de Catalunya.

* Corresponding author.

*Email addresses:* conrado-at-lsi.upc.es (Conrado Martínez), hproding-at-sun.ac.za (Helmut Prodinger).

[1] Supported by the Spanish Min. of Science and Technology project TIN2006-11345 (ALINEX).

[2] Supported by the South African Science Foundation NRF, grant 2053748.

algorithm [2]. This kind of analysis is useful whenever we encounter a situation where there is some associated cost $C(i, j, \ell)$ to move element $i$ from position $j$ to position $\ell$.

We will consider here two parameters of interest: 1) the number of moves $M_{n,i}$ of element $i$ when we sort an array of size $n$, and 2) the (accumulated) displacement $D_{n,i}$ of element $i$. The first random variable corresponds to the situation where $C(i, j, \ell) = 1$ whenever $j \neq \ell$ and $C(i, j, \ell) = 0$ otherwise. For the second random variable, we take into account the number of positions that the element travels each time it is moved; thus, $C(i, j, \ell) = |j - \ell|$. Section 2 is devoted to the analysis of $M_{n,i}$ and Section 3 to the analysis of $D_{n,i}$. Although the same techniques could be used to investigate the variance and higher order moments, we will not do so here, because the computations involved are too cumbersome.

We use in this paper fairly standard tools in the analysis of algorithms such as *probability generating functions* (see, for instance, [7]). As we will see in later sections, the analysis of data moves in quicksort involves the so called *quickselect recurrence*. In its general standard form it reads

$$f_{n,i} = a_{n,i} + \frac{1}{n} \sum_{1 \leq k < i} f_{n-k,i-k} + \frac{1}{n} \sum_{i < k \leq n} f_{k-1,i}, \quad 1 \leq i \leq n, \tag{1}$$

for some given *toll* function $a_{n,i}$. As its name suggests, this type of recurrence appears in the analysis of *quickselect* [1], a variant of quicksort where we only need to select the $i$-th smallest element out of $n$ rather than sorting the whole array.

In a recent paper, M. Kuba has provided the general solution to the recurrence above; we reproduce here his main result for the reader's convenience, as we shall use it frequently in the sequel [3].

**Theorem 1 (Kuba [5])** *The value $f_{n,i}$ defined by (1) with arbitrary fixed values $a_{n,i}$, $1 \leq i \leq n$, is given by*

$$f_{n,i} = a_{1,1} + \sum_{k=n+2-j}^{n} A(k, k-n+j) + \sum_{k=2}^{n+1-j} \frac{k a_{k,1} - (k-1) a_{k-1,1}}{k},$$

[3] Some of Kuba's original formulas read incorrectly due to improper formatting; see the author's homepage for the corrected versions [4] given here in Theorem 1.

*where* $A(n,i)$ *is given by*

$$A(n,i) = \sum_{k=j+1}^{n} \frac{ka_{k,i} - (k-1)a_{k-1,i-1} - (k-1)a_{k-1,i} + (k-2)a_{k-2,i-1}}{k} + \frac{ia_{i,i} - (i-1)a_{i-1,i-1}}{i}.$$

We review now how quicksort works, and in particular, its partitioning procedure. Actually, there are several different partitioning schemes, each one with its virtues and drawbacks. Each one would require an independent analysis, as they essentially differ in the way they move the data to reorganize the array around the pivot. We will concentrate in the standard scheme [6], for which we analyze the number of moves (Section 2) and displacement (Section 3); we also consider the behavior of the number of moves for a symmetric variant of the partitioning scheme (Section 4) to exemplify how we could analyse different partitioning schemes using the same basic set of tools.

Quicksort sorts the subarray $A[l..u]$ by reorganizing its contents of around a *pivot* element $p = A[l]$; upon exit of the partitioning procedure $A[k] = p$, all elements in $A[l..k-1]$ are smaller than or equal to $p$ and all elements in $A[k+1..u]$ are greater than or equal to $p$. Hence, the pivot has been brought to its correct position, and the algorithms recursively calls itself on the subarrays $A[l..k-1]$ and $A[k+1..u]$ to the left and to the right of the pivot, respectively (see Algorithm 1).

**Algorithm 1** The quicksort algorithm.

```
procedure QUICKSORT(A, l, u)
    if l ≥ u then return ▷ Nothing needs to be done
    end if
    ▷ p = A[l]
    PARTITION(A, l, u, k)
        ▷ ∀i : l ≤ i < k : A[i] ≤ p, A[k] = p, and ∀i : k < i ≤ u : p ≤ A[i]
    QUICKSORT(A, l, k − 1)
    QUICKSORT(A, k + 1, u)
end procedure
```

The partition procedure scans the current subarray from both ends. The pivot element $p$ is located at $A[l]$. At any intermediate stage $A[l+1..i-1]$ contains elements $\le p$ and $A[j+1..u]$ contains elements $\ge p$. The two internal loops scan the subarray from left to right and from right to left until some $A[i] > p$ and $A[j] < p$ have been found (or the scanning has finished). If elements $A[i] > p$ and $A[j] < p$ have been found then they are swapped and we resume the scanning of the subarray (see Algorithm 2).

**Algorithm 2** Partition $A[l..u]$ around the pivot at $A[l]$ and return the final position $k$ of the pivot.

```
procedure PARTITION(A, l, u, k)
    p ← A[l]
    i ← l; j ← u + 1
    loop
        repeat i ← i + 1
        until A[i] ≥ p
        repeat j ← j − 1
        until A[j] ≤ p
        if i ≥ j then
            break
        end if
        A[i] ↔ A[j]
    end loop
    A[l] ↔ A[j]
    k ← j
end procedure
```

## 2 The number of moves

For the analysis below and the rest of the paper, we will assume w.l.o.g. that the array to be sorted contains a random permutation of $\{1, \ldots, n\}$. Assume that the pivot is the $k$-th element. Consider now some element $i < k$. If $i$ belongs to $A[2..k-1]$ prior to the partitioning it will not move. On the contrary, if $i$ were initially located at any position of $A[k..n]$ then it will be moved to stay to the left of the pivot. Thus, with probability $(k-2)/(n-1)$ the element doesn't move and with probability $(n+1-k)/(n-1)$ it does.

Similarly, if $i > k$ then it doesn't move if it were initially located within $A[k+1..n]$ —this happens with probability $(n-k)/(n-1)$—, whereas it will be moved whenever it were located in $A[2..k]$, hence with probability $(k-1)/(n-1)$.

Finally, if $i = k$ then it will be moved once to its final position; we count as a move the degenerate case where $i = 1$, since the partitioning algorithm performs a redundant exchange in this case.

Once the (eventual) move of element $i$ has been taken into account for the current partitioning stage, we keep track of the subsequent moves of the element $i$ while sorting the left subarray of size $k-1$ if $i < k$, or while sorting the right subarray of size $n-k$ if $i > k$ (but we have to track down the whereabouts of the element $i-k$ there).

From the discussion above, the following theorem is immediate.

**Theorem 2** *The probability generating function $M_{n,i}(v)$ of the number of moves of element $i$ in a random permutation of $\{1, 2, \ldots, n\}$, satisfies the recursion*

$$M_{n,i}(v) = \frac{1}{n} \sum_{1 \le k < i} \left( \frac{k-1}{n-1} v + \frac{n-k}{n-1} \right) M_{n-k,i-k}(v) + \frac{1}{n} \sum_{i < k \le n} \left( \frac{k-2}{n-1} + \frac{n-k+1}{n-1} v \right) M_{k-1,i}(v) + \frac{v}{n}$$

*for $n \ge 2$ and $1 \le i \le n$; $M_{1,1}(v) = v$.*

**Corollary 3** *The expected number of moves $\mu_{n,i} = M'_{n,i}(1)$ satisfies*

$$\mu_{n,i} = \frac{1}{n} \sum_{1 \le k < i} \mu_{n-k,i-k} + \frac{1}{n} \sum_{i < k \le n} \mu_{k-1,i} + \frac{1}{n} + \frac{(i-1)(i-2)}{2n(n-1)} + \frac{(n+1-i)(n-i)}{2n(n-1)}, \quad n > 1, 1 \le i \le n,$$

*with $\mu_{1,1} = 1$.*

To solve the recurrence above we use Theorem 1. Here, $a_{n,i} = \frac{1}{n} + \frac{(i-1)(i-2)}{2n(n-1)} + \frac{(n+1-i)(n-i)}{2n(n-1)}$.

**Theorem 4** *For all $n > 1$, $1 \le i \le n$,*

$$\mu_{n,i} = \frac{1}{3} H_n + \frac{1}{6} H_i + \frac{1}{6} H_{n+1-i} + \frac{1}{6} + \frac{1}{3i} - \frac{(i-1)^2}{3n} + \frac{(i-1)(i-2)}{3(n-1)} + \frac{1}{12} [\![ i = 1 ]\!] - \frac{1}{12} [\![ i = n ]\!],$$

*where $H_n = \sum_{1 \le k \le n} \frac{1}{k}$ is the $n$-th harmonic number and $[\![ P ]\!] = 1$ if $P$ is true and $[\![ P ]\!] = 0$ otherwise.*

A few additional computations with the formula above yield simple asymptotic estimates for interesting special cases.

**Corollary 5** *For fixed $i \ge 1$, as $n \to \infty$,*

$$\mu_{n,i} = \frac{1}{2} \ln n + \frac{1}{6} H_i + \frac{\gamma}{2} + \frac{1}{3i} + \frac{1}{6} + O(n^{-1}),$$

*where $\gamma = 0.577215\ldots$ is Euler's gamma constant. Furthermore, if $i > 1$, $\mu_{n,n+1-i} = \mu_{n,i} - \frac{1}{3i} + O(n^{-1})$.*

*For $i = \alpha n + o(n)$, $0 < \alpha < 1$, we have*

$$\mu_{n,i} = \frac{2}{3} \ln n + \frac{1}{6} + \frac{1}{6} \ln \alpha + \frac{1}{6} \ln(1-\alpha) - \frac{\alpha(1-\alpha)}{3} + \frac{2}{3} \gamma + O(n^{-1}).$$

*The global minimum of* $\mu_{n,i}$ *occurs at* $i = n$. *The maximum of* $\mu_{n,\alpha\cdot n}$ *occurs close to the median* ($\alpha = 1/2$)*, actually at*

$$\alpha^* = \frac{1}{2} - \frac{2}{n} - \frac{39}{n^2} - \frac{582}{n^3} - \frac{8604}{n^4} - \frac{121168}{n^5} + O(n^{-6}),$$

*with* $\mu_{n,\alpha^* n} = \frac{2}{3}\ln n + \frac{1}{12} - \frac{1}{3}\ln 2 + \frac{2}{3}\gamma + O(n^{-1})$.

Another quantity of interest is the cumulated number of moves. By linearity, its expected value is the sum of the $\mu_{n,i}$'s.

**Corollary 6** *For* $n \geq 2$, *the total number of moves is given by*

$$\overline{\mu}_n = \sum_{i=1}^{n} \mu_{n,i} = \frac{2}{3}(n+1)H_n - \frac{4n+1}{18}.$$

As indicated already in the Introduction, we refrain from going further, since the variance with a simpler toll function has proved to be quite formidable [3].

## 3 Displacement

Now we measure the "distances" that the individuals travel: instead of just counting how many times some element $i$ has moved," we record the (cumulative) distance of where $i$ was and where it is after each iteration. We do here a case analysis as in the previous section. Suppose $i > k$. Then it will be moved if it were at some source position $j$ between 2 and $k$; it will land at some target position $\ell$ between $k+1$ and $n$. The displacement at that particular stage is hence $\ell - j$. Now, the probability that $i$ has to move is $(k-1)/(n-k)$. Conditioned on the event that $i$ has to move, any source position $j$ between 2 and $k$ is equally likely, i.e., has probability $1/(k-1)$. Analogously, given that $i$ is kicked out from its source position $j$, any target position $\ell$ between $k+1$ and $n$ has identical probability $1/(n-k)$. So when we consider the PGF for $D_{n,i}$ we will have a contribution of the form

$$\left(\left(\sum_{j=2}^{k} \sum_{\ell=k+1}^{n} \frac{v^{\ell-j}}{(n-1)(n-k)}\right) + \frac{n-k}{n-1}\right) D_{n-k,i-k}(v),$$

for each possible pivot $k < i$. The other cases, when $i < k$ and when $i = k$ are dealt with similarly; in particular, if $i = k$ then the element has to be moved from the first position (the original position of the pivot) to position $i$. Summing up everything, we have the following theorem.

**Theorem 7** *The probability generating function* $D_{n,i}(v)$ *of the displacement* *of element $i$ in a random permutation of $\{1,2,\ldots,n\}$, satisfies the recursion*

$$D_{n,i}(v) = \frac{1}{n}\sum_{1\le k<i}\left(\sum_{j=2}^{k}\sum_{\ell=k+1}^{n}\frac{v^{\ell-j}}{(n-1)(n-k)}+\frac{n-k}{n-1}\right)D_{n-k,i-k}(v)$$
$$+\frac{1}{n}\sum_{i<k\le n}\left(\frac{k-2}{n-1}+\sum_{j=1}^{k-1}\sum_{\ell=k}^{n}\frac{v^{\ell-j}}{(n-1)(k-1)}\right)D_{k-1,i}(v)+\frac{v^{i-1}}{n}$$

*for $n\ge 2$ and $1\le i\le n$; $D_{1,1}(v)=1$.*

Then we follow the path already traced when analyzing the number of moves: 1) obtain a recursion for expected values by differentiating the recursion for PGFs and setting $v=1$; 2) solve the recursion using the general result by Kuba; 3) analyze some special cases of interest and the total displacement (the sum of all individual displacements).

**Corollary 8** *The expected displacement $\delta_{n,i}=D'_{n,i}(1)$ satisfies*

$$\delta_{n,i}=\frac{1}{n}\sum_{1\le k<i}\delta_{n-k,i-k}+\frac{1}{n}\sum_{i<k\le n}\delta_{k-1,i}+\frac{(i-1)(i-2)}{4n}+\frac{(n-i)(n+1-i)}{4(n-1)}+\frac{i-1}{n}$$

*with $\delta_{1,1}=0$.*

**Theorem 9** *For all $n>1$ and $1\le i\le n$,*

$$\delta_{n,i}=\frac{n}{2}+\frac{1}{12}H_n-\frac{1}{12}H_{i-1}-\frac{1}{3}H_{n+1-i}+\frac{5}{24}-\frac{(i-1)^2}{12n}+\frac{(i-1)(i-2)}{12(n-1)}$$
$$+\frac{1}{6}[\![i=1]\!]+\frac{1}{8}[\![i=n]\!].$$

**Corollary 10** *For fixed $i\ge 1$, as $n\to\infty$,*

$$\delta_{n,i}=\frac{n}{2}-\frac{1}{4}\ln n+O(1),$$
$$\delta_{n,n+1-i}=\frac{n}{2}+O(1).$$

*For $i=\alpha n+o(n)$, $0<\alpha<1$,*

$$\delta_{n,i}=\frac{n}{2}-\frac{1}{3}\ln n-\frac{1}{12}\ln\alpha-\frac{1}{3}\ln(1-\alpha)-\frac{\alpha}{12}+\frac{\alpha^2}{12}+\frac{5}{24}+O(n^{-1}).$$

*The maximum of $\delta_{i,n}$ occurs at $i=n$; there $\delta_{n,n}=n/2$. The minimum of $\delta_{n,\alpha n}$ occurs at*

$$\alpha^*=\frac{5}{4}-\frac{\sqrt{17}}{4}+\left(\frac{5}{8}+\frac{5}{136}\sqrt{17}\right)\frac{1}{n}+\left(-\frac{33}{256}+\frac{821}{221952}\sqrt{17}\right)\frac{1}{n^2}$$
$$+\left(\frac{981}{4096}+\frac{4864631}{60370944}\sqrt{17}\right)\frac{1}{n^3}+O(n^{-4})=0.219223594\ldots+O(n^{-1}),$$

*with*

$$\delta_{n,\alpha^* n} = \frac{n}{2} - \frac{1}{3}\ln n - \frac{1}{12}\ln\left(\frac{5}{4} - \frac{\sqrt{17}}{4}\right) - \frac{1}{3}\ln\left(-\frac{1}{4} + \frac{\sqrt{17}}{4}\right) + \frac{31}{96} - \frac{\sqrt{17}}{32} - \frac{\gamma}{3} + O(n^{-1}).$$

**Corollary 11** *For $n \geq 2$, the total displacement is given by*

$$\bar{\delta}_n = \sum_{i=1}^{n} \delta_{n,i} = \frac{n(9n+11)}{18} - \frac{1}{3}(n+1)H_n + \frac{5}{18}.$$

**Remark.** The average displacement of element $i$ in a random permutation is

$$\frac{1}{n}\sum_{k=1}^{n} |i-k| = \frac{i(i-1)}{2n} + \frac{(n+1-i)(n-i)}{2n}.$$

This is for $i = \alpha n + o(n)$ asymptotic to $\frac{n}{2}(\alpha^2 + (1-\alpha)^2)$.

## 4 Symmetric partitioning

To break the asymmetry of taking the *first* element as pivot, we choose a *random* location and take the element there as the pivot. Pivot $k$ is in location $\ell$ and the particular element $i$ in location $j$. If we repeat the case analysis of Section 2, we arrive at

$$\begin{aligned} M_{n,i}(v) = \sum_{1 \leq k < i} &\left[\sum_{k<j\leq n} \frac{1}{n(n-1)} + \sum_{1\leq j<k} \frac{v}{n(n-1)}\right] M_{n-k,i-k}(v) \\ + \sum_{i<k\leq n} &\left[\sum_{k<j\leq n} \frac{v}{n(n-1)} + \sum_{1\leq j<k} \frac{1}{n(n-1)}\right] M_{k-1,i}(v) + \frac{v}{n}, \end{aligned}$$

which simplifies to

$$\begin{aligned} M_{n,i}(v) &= \frac{1}{n}\sum_{1\leq k<i} \left(\frac{k-1}{n-1}v + \frac{n-k}{n-1}\right) M_{n-k,i-k}(v) \\ &+ \frac{1}{n}\sum_{i<k\leq n} \left(\frac{k-1}{n-1} + \frac{n-k}{n-1}v\right) M_{k-1,i}(v) + \frac{v}{n}. \end{aligned}$$

Hence, we get again the quickselect-type recurrence for $\mu_{n,i} = M'_{n,i}(1)$, but this time the toll function is

$$a_{n,i} = \frac{1}{2} + \frac{1}{n-1} - \frac{i(n+1-i)}{n(n-1)}.$$

Following the same steps as in previous sections we arrive at the solution

$$\mu_{n,i} = \frac{1}{3}H_n + \frac{1}{6}H_i + \frac{1}{6}H_{n+1-i} - \frac{1}{6} - \frac{1}{6n} + \frac{1}{3}\frac{1}{n-1} + \frac{1}{6}\left(\frac{1}{i} + \frac{1}{n+1-i}\right) - \frac{1}{3}\frac{i(n+1-i)}{n(n-1)} + \frac{1}{12}[\![i=1]\!] + \frac{1}{12}[\![i=n]\!],$$

which is clearly symmetric: $\mu_{n,i} = \mu_{n,n+1-i}$. The difference between the number of moves for this symmetric partitioning, and that for the standard partitioning in Section 2 is asymptotically negligible, actually it is about $\frac{1}{3}$. Hence, we have the same asymptotic estimates for this $\mu_{n,i}$ as we had in Section 2. Namely, for fixed $i \geq 1$, $\mu_{n,i} = \frac{1}{2}\ln n + O(1)$, and for $i = \alpha n + o(n)$, $\mu_{n,i} = \frac{2}{3}\ln n + O(1)$.

Last but not least, the total number of moves follows by simple summation:

$$\overline{\mu}_n = \sum_{i=1}^{n} \mu_{n,i} = \frac{2}{3}(n+1)H_n - \frac{5n-1}{9}.$$

which is basically the total number of moves in the standard scheme minus $n/3$.

Also, if we consider displacements with this symmetric variant of partitioning, we have for the expected value the quickselect recurrence with toll function

$$a_{n,i} = \frac{(i-1)^2}{4(n-1)} + \frac{(n-i)^2}{4(n-1)} + \frac{1}{2n} - \frac{(i-1)(n-i)}{2n(n-1)}, \quad n > 1, 1 \leq i \leq n,$$

and $a_{1,1} = 0$.

Using Theorem 1 we get

$$\delta_{n,i} = \frac{n}{2} + \frac{1}{3}H_n - \frac{5}{6}H_i - \frac{5}{6}H_{n+1-i} + 1 + \frac{1}{6}\left(\frac{1}{i} + \frac{1}{n+1-i}\right) - \frac{1}{3}\frac{i(n+1-i)}{n(n-1)} + \frac{1}{3}\frac{1}{n-1} - \frac{1}{6n} - \frac{1}{12}[\![i=1]\!] - \frac{1}{12}[\![i=n]\!].$$

Again, this formula is obviously symmetric ($\delta_{n,i} = \delta_{n,n+1-i}$), and attains its global minimum at $i = \lfloor n/2 \rfloor$.

When $i$ is fixed and $i > 1$ then

$$\delta_{n,i} = \delta_{n,n+1-i} = \frac{n}{2} - \frac{1}{2}\ln n + 1 - \frac{1}{2}\gamma - \frac{5}{6}H_i + O(n^{-1}),$$

and for $i = \alpha n + o(n)$, $0 < \alpha < 1$,

$$\delta_{n,i} = \frac{n}{2} - \frac{4}{3}\ln n + 1 - \frac{4}{3}\gamma - \frac{1}{3}\alpha(\alpha+1) - \frac{5}{6}\ln\alpha - \frac{5}{6}\ln(1-\alpha) + O(n^{-1}).$$

The average displacement under this partitioning scheme is always smaller than for the non-symmetric variant, with the difference ranging from $\frac{1}{4}\ln n + O(1)$ when $i$ is fixed to $\ln n + O(1)$ when $i = \alpha n + o(n)$ for some $\alpha$, $0 < \alpha < 1$.

Summing up for all $i$ we get the average total displacement, namely,

$$\overline{\delta}_n = \sum_{1\le i\le n} \delta_{n,i} = \frac{n(9n+47)}{18} - \frac{4}{3}(n+1)H_n - \frac{1}{18},$$

which improves the average total displacement of the standard partition by $n\ln n + O(n)$.

## References


[1] C.A.R. Hoare. Find (Algorithm 65). *Communications of the ACM*, 4:321–322, 1961.

[2] C.A.R. Hoare. Quicksort. *Computer Journal*, 5:10–15, 1962.

[3] P. Kirschenhofer and H. Prodinger. Comparisons in Hoare's Find algorithm. *Combinatorics, Probability and Computing*, 7:111–120, 1998.

[4] M. Kuba. Addendum: On quickselect, partial sorting and multiple quickselect. Available at `http://www.dmg.tuwien.ac.at/kuba/`, 2006.

[5] M. Kuba. On quickselect, partial sorting and multiple quickselect. *Information Processing Letters*, 99(5):181–186, 2006.

[6] R. Sedgewick. *Quicksort.* Garland, New York, 1978.

[7] R. Sedgewick and Ph. Flajolet. *An Introduction to the Analysis of Algorithms.* Addison-Wesley, Reading, Mass., 1996.