

The Analysis of Multiple Quickselect

Helmut Prodinger

Technical University of Vienna

June 30, 1997

[summary by Philippe Flajolet]

Abstract

Quickselect is a version of Quicksort that makes it possible to find efficiently any element in an unsorted file given its rank. “Multiple Quickselect” is designed to find simultaneously a collection of elements, also specified by their ranks. This talk shows how to analyse Multiple Quickselect when the underlying permutation is random and the collection of ranks is a random p -subset (p fixed). The analysis provides a nice illustration of the use of multivariate generating functions.

1. Algorithms

The principle of *Quicksort* is of course extremely well known. Given an array $T[1 \dots n]$ of numbers to be sorted, choose a “pivot”, say $T[1]$, then partition (by a linear scan) the original array into the two subarrays, $T_{<}$ and $T_{>}$, formed by elements that are respectively smaller and larger than the pivot, and proceed recursively. The algorithm was first proposed by Hoare in 1962. Its characteristics are fairly well understood: the algorithm sorts in place (it uses $O(1)$ auxiliary memory in addition to the recursion stack), its average number of comparisons is $\sim 2n \log n$, while the implementation constants are especially low. For these reasons, Quicksort is a method of choice amongst sorting algorithms, and it has been adopted as the basis of the Unix `sort` command. Note however that it is still unknown whether the limit distribution of the number of comparisons that has been proved to exist can be characterized in terms of standard special functions of analysis. (Existence proofs comprise the martingale argument of Régnier, the moment method of Hennequin, and the contraction metrics approach of Rösler; see [4] and references therein.)

Quickselect is a simplified version of Quicksort adapted so as to locate the j th ranked element in a file. In that case, it suffices to effect one recursive descent in one of the two subfiles $T_{<}, T_{>}$ created by the basic partitioning stage of Quicksort. Knuth’s book [2] provides the analysis of the two parameters of interest, the average number of passes (recursive calls) $P[n; j]$ and the average number of comparisons $C[n; j]$,

$$P[n; j] = H_j + H_{n+1-j} - 1,$$

$$C[n; j] = 2 \left(n + 3 + (n + 1)H_n - (j + 2)H_j - (n + 3 - j)H_{n+1-j} \right)$$

where $H_n = 1 + \frac{1}{2} + \dots + \frac{1}{n}$ denotes the n th harmonic number. In particular, the mean number of comparisons is $O(n)$, uniformly for all j . From the complexity standpoint, the algorithm thus has the same type of cost as a fixed number of scans of the input file—a highly appealing feature. (Contrast the simplicity of Quickselect with algorithms like median-finding that are constructed specifically to achieve linear complexity in the worst-case!)

Multiple Quickselect is an extension of Hoare's idea that works as follows: Assume that we simultaneously search for p elements with ranks $J = (j_1, j_2, \dots, j_p)$ where $1 \leq j_1 < j_2 < \dots < j_p \leq n$ is a fixed set of p values. Then, depending on the interval determined by j_1, \dots, j_p in which the pivot element falls, we continue recursively on both subfiles $T_<, T_>$, but with smaller J -index sets. The principle is quite simple and a nice description of the algorithm can be found in [3]. Notice that Multiple Quickselect can be used for instance to determine quantiles of distributions efficiently.

In a previous work, Prodinger [7] has given explicit formulæ for both the average number of passes $P[n; j_1, \dots, j_p]$ and the average number of comparisons, $C[n; j_1, \dots, j_p]$. As a consequence, the so-called “*grand averages*”,

$$\mathcal{P}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1 \leq j_1 < j_2 < \dots < j_p \leq n} P[n; j_1, \dots, j_p],$$

$$\mathcal{C}_{n,p} = \frac{1}{\binom{n}{p}} \sum_{1 \leq j_1 < j_2 < \dots < j_p \leq n} C[n; j_1, \dots, j_p]$$

have been determined (see also [3]). However, the approach used in [7] was a mixture of guessing and proofs by induction.

The present paper by Panholzer and Prodinger [6] develops a direct generating function approach to the analysis of the grand averages. This gives access to variances that were out of reach with the old method.

2. Moments

In both instances, namely number of passes and of comparisons, the problem is modelled by a trivariate generating function $\Phi(z, u, v)$, where the coefficient $n! [z^n u^p v^m] \Phi$ is the number of permutations of $\{1, 2, \dots, n\}$ and of subsets of p elements of $\{1, 2, \dots, n\}$ such that the parameter of interest has value m . Under the random permutation model, the splitting probabilities of Quicksort and Quickselect are given by

$$\Pr\{K = k\} = \frac{1}{n},$$

where the random variable $K \in [1, n]$ denotes the rank of the pivot. The model is thus isomorphic to that of binary search trees (BST's) or heap-ordered trees [2, 4].

Number of passes. The trivariate generating function (GF) where z records the size n of the input permutation, v records the number of passes (recursive calls), and u records the number of elements being simultaneously selected satisfies

$$(1) \quad \Phi'(z, u, v) = v(1 + u)\Phi^2(z, u, v) + \frac{1 - v}{(1 - z)^2}$$

with $\Phi(0, u, v) = 1$ and Φ' denotes a partial derivative with respect to z . The relation (1) reads off almost directly from the problem. We may also view it as expressing the size of a generalized common ancestor tree in binary search trees.

As is usual in distributional analyses of additive parameters on BST's, we have a Riccati equation; see for instance, the analysis of patterns in BST's in [1]. Such equations are systematically linearized by a change of variable of the form $\Phi(z) = a(z)W'(z)/W(z)$. Here,

$$(2) \quad \Phi(z, u, v) = \frac{\Omega + 1 - 2v + (1 - z)^\Omega(\Omega - 1 + 2v)}{\left(\Omega + 1 - 2v(1 + u) + (1 - z)^\Omega(\Omega - 1 + 2v(1 + u))\right)(1 - z)}$$

$$\Omega = \sqrt{1 - 4v(1+u)(1-v)}.$$

(This is well within the automatic capabilities of Maple.) By differentiating with respect to v and expanding, we obtain a result of [7].

Theorem 1. For $p \geq 1$ the average number $\mathcal{P}_{n,p}$ of passes employed by Multiple Quickselect in order to search for p random elements is

$$(3) \quad \mathcal{P}_{n,p} = (H_{n+1} - H_p) \frac{2p(n+1)^2}{(n+2-p)(n+1-p)} - \frac{n(2p-1)+p}{n+2-p}.$$

A particular case is the basic Quickselect algorithm ($p = 1$), for which

$$\mathcal{P}_{n,1} = 2 \left(1 + \frac{1}{n}\right) H_n - 3.$$

By taking second derivatives, we have access to the variance. A careful expansion guided by educated guesses and patience then gives explicit expressions. We state the result from [6] as it is a good indication of the intricacies involved.

Theorem 2. The variance of the number of passes when searching for a random set of $p \geq 2$ elements with Multiple Quickselect is given by

$$\begin{aligned} V_{n,p} = & -\frac{4p(n+1)^2\Psi_1}{(n+4-p)(n+3-p)(n+2-p)^2(n+1-p)^2} (H_{n+1} - H_p)^2 \\ & + \frac{2(n+1)^2\Psi_2}{(n+4-p)(n+3-p)(n+2-p)^2(n+1-p)} (H_{n+1} - H_p) \\ & - \frac{4p(n+2)(n+1)^2(pn+p+2)}{(n+4-p)(n+3-p)(n+2-p)(n+1-p)} (H_{n+1}^{(2)} - H_p^{(2)}) \\ & + \frac{2(n+1)^2\Psi_3}{(n+4-p)(n+3-p)(n+2-p)^2} \end{aligned}$$

$$\Psi_1 = (3p-2)n^3 - 2(p^2-9p+5)n^2 - (p^3+5p^2-33p+16)n - p^3 - 5p^2 + 20p - 8$$

$$\Psi_2 = pn^3 + (11p^2-13p+8)n^2 - (9p^3-54p^2+62p-32)n - 3p^4 - p^3 + 38p^2 - 56p + 32$$

$$\Psi_3 = (2p-1)n^2 - 2(5p^2-15p+8)n + 8p^3 - 45p^2 + 72p - 32.$$

For $p = 1$, the variance simplifies to

$$V_{n,1} = -4\frac{n+1}{n^2}H_{n+1}^2 + \frac{2(n+4)(n+1)}{n^2}H_{n+1} - 4\frac{n+1}{n}H_{n+1}^{(2)} + \frac{2(2n^2-n-2)}{n^2}.$$

Number of comparisons. The process for moment calculations is similar, only the expressions become a little more complicated. The fundamental functional equation is now of the difference-differential type,

$$\Phi'(z, u, v) = (1+u)\Phi^2(zv, u, v) + \frac{1}{(1-z)^2} - \frac{1}{(1-zv)^2},$$

with $\Phi(0, u, v) = 1$. Contrary to the previous case, this equation is not known to have a closed form solution. It is however still possible to “pump” GF’s of moments by differentiation. We must omit details here and simply state the average-case result while referring to [6] for a daunting variance computation that involves the dilogarithm $\text{Li}_2(z) := \sum z^n/n^2$.

Theorem 3. For $p \geq 1$ the average number $C_{n,p}$ of comparisons to search for p random elements with Multiple Quickselect is given by

$$C_{n,p} = -\frac{2p(n+1)(4n-p+5)}{(n+2-p)(n+1-p)}H_{n+1} + \frac{2(n+1)^2(n+2p+2)}{(n+2-p)(n+1-p)}H_p + \frac{n^2 + (5p-1)n + 3p}{n+2-p}.$$

In particular for a basic (single element) Quickselect and a full sort, we get back the classical results,

$$C_{n,1} = 3n - 8 \left(1 + \frac{1}{n}\right) H_n + 13, \quad C_{n,n} = 2(n+1)H_n - 4n.$$

3. Distributions

The limiting distribution of Quicksort has been under attack for about 20 years. For basic Quickselect, the problem is in a way simpler since the recursion structure is a linear one. Here is what is known at present:

- The number of passes of multiple quickselect (p fixed) is asymptotically normal. This observation results rather directly from singularity perturbation methods applied to the explicit form (2) of the trivariate GF (Flajolet & Prodinger 1997, unpublished).
- The number of comparisons of basic (single) Quickselect has been studied by Mahmoud *et al.* [5] who determined explicitly the characteristic function of the limit law. The main result of [5] entails that this limit has the same distribution as the sum of a Poisson number of independent random variables with an elementary density.

The second result suggests that there is an interesting class of limit distributions for comparison costs when p is a fixed integer $p \geq 2$.

References

- [1] Flajolet (Philippe), Gourdon (Xavier), and Martínez (Conrado). – *Patterns in random binary search trees.* – Research Report n° 2997, Institut National de Recherche en Informatique et en Automatique, October 1996. 23 pages. To appear in *Random Structures & Algorithms*.
- [2] Knuth (Donald E.). – *The Art of Computer Programming.* – Addison-Wesley, 1973, vol. 3: Sorting and Searching.
- [3] Lent (Janice) and Mahmoud (Hosam M.). – Average-case analysis of multiple Quickselect: an algorithm for finding order statistics. *Statistics & Probability Letters*, vol. 28, n° 4, 1996, pp. 299–310.
- [4] Mahmoud (Hosam M.). – *Evolution of Random Search Trees.* – John Wiley & Sons Inc., New York, 1992, *Wiley-Interscience Series in Discrete Mathematics and Optimization*, xii+324p.
- [5] Mahmoud (Hosam M.), Modarres (Reza), and Smythe (Robert T.). – Analysis of QUICKSELECT: an algorithm for order statistics. *RAIRO Theoretical Informatics and Applications*, vol. 29, n° 4, 1995, pp. 255–276.
- [6] Panholzer (Alois) and Prodinger (Helmut). – A generating functions approach for the analysis of grand averages for multiple quickselect. – Preprint, 1997.
- [7] Prodinger (H.). – Multiple quickselect — Hoare’s find algorithm for several elements. *Information Processing Letters*, vol. 56, 1995, pp. 123–129.