Algorithms Seminar 2001–2002,
F. Chyzak (ed.), INRIA, (2003), pp. 57–62.

Available online at the URL
http://algo.inria.fr/seminars/.

Everything You Always Wanted to Know about Quicksort, but Were Afraid to Ask

Marianne Durand
Projet Algorithmes, Inria Rocquencourt (France)

November 5, 2001

Summary by Michel Nguyễn-Thế

Abstract

The algorithm Quicksort was invented by Hoare in 1960. Numerous improvements have been suggested since then, like optimization of the choice of the pivot or simultaneous use of several pivots or also hybrid methods. Different parameters like the cost of comparisons, the size or the height of the associated binary search tree have been studied for Quicksort and its variants. We present here the principal methods used to get the mean, the variance, and the nature or at least a few properties of the limit laws of these parameters.

1. Description of the Algorithm and of a Few Variants

1.1. **Quicksort.** The procedure Quicksort takes as arguments an array A of n elements and two integers First and Last representing indices of elements of the array. The algorithm runs as follows: if First $<$ Last then:

1. Choose a pivot in the array (e.g., A[First]).
2. Partition the elements in the subarray A[First] ... A[Last] so that the pivot value is in place (let PivotIndex be its position then).
3. Apply Quicksort to the first subarray A[First] ... A[PivotIndex $-$ 1].
4. Apply Quicksort to the second subarray A[PivotIndex + 1] ... A[Last].

1.2. **Variants.** In step 1, the pivot is chosen in a fixed manner. It is possible to use a strategy to choose the pivot to improve the efficiency of the algorithm. By choosing the pivot randomly, we can wipe out the possible bias of the data we want to sort. The pivot is all the more efficient if it cuts the array in two arrays of similar size. With this aim in view, the Quicksort with median of $2t+1$ consists in picking out $2t+1$ elements randomly in the array to sort, where t is a fixed integer, and to choose as pivot the $(t+1)$th element among the picked elements. Martínez and Roura [7] even analysed the situation with a sample size depending on n, and obtained that the optimal sample size to minimize the average total cost of Quicksort, including comparisons and exchanges, is $s = a\sqrt{n} + o(\sqrt{n})$, for some constant a. Quicksort with 3–3 median consists in picking 3 samples of the array, each of 3 elements. We take the median element of each sample, so that we are left with three elements, of which we take again the median element, that we finally choose as pivot. This strategy can be furthered in choosing $m-1$ pivots or medians, among $m(t+1)-1$ elements, instead of one only, that leaves us with sorting recursively m subarrays instead of two only.

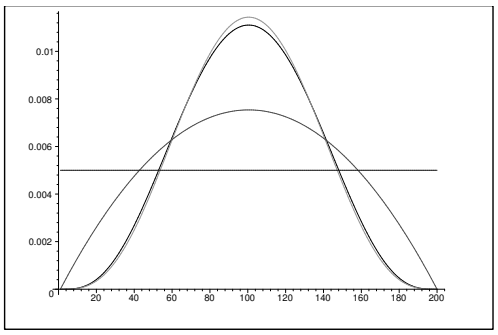

FIGURE 1. Probability of choice of pivot.

1.3. **Parameters.** The parameters of interest are: the cost in number of comparisons, that is the internal path length of the associated search tree; the size of the associated m-ary tree; the profile of the tree. Notice that the size of a binary tree is the number of internal nodes, and that the size of an m-ary tree for $m \geq 3$ is the number of both internal and external nodes. We can generally compute the first moments or cumulants of these parameters, especially the mean and the variance, by using generating functions. With the aid of various other tools, it is possible to show the existence of a limit law for the cost of Quicksort and to derive some properties of this law. For the other parameters, the knowledge of the moments sometimes turns out to be sufficient to establish a Gaussian limit.

2. Moments of Internal Path Length

2.1. **Expectation.** Recall that m is the arity of the tree. The cost expectation f_n of Quicksort and its variants is given by the recurrence relation

$$f_n = t_n + \sum_{k=1}^{n} \mathbf{P}(\text{PivotIndex} = k)\,(f_{k-1} + f_{n-k}), \tag{1}$$

which can be rewritten into the equation $\mathcal{L}\big(f(z)\big) = (1-z)^{-\beta}$, where $f(z) = \sum_n z^n$, and the operator $\mathcal{L}$ is of the form $\mathcal{L}(y) = a_m(1-z)^m y^{(m)} + \cdots + a_0 y$. There exists a polynomial $I(\alpha)$, called the index polynomial, such that $\mathcal{L}\big((1-z)^{-\alpha}\big) = I(\alpha)(1-z)^{-\alpha}$. The solutions of $\mathcal{L}(y) = 0$ are given by $(1-z)^{-\alpha}\log^k \frac{1}{1-z}$ with $I(\alpha) = 0$ and k smaller than the order of multiplicity of root α. Given the initial conditions and the particular solution $I^{(r)}(\beta)(1-z)^{-\beta}\big(\log \frac{1}{1-z}\big)^r$, where r is the order of β as root of I (r can be zero), it is then easy to get the right solution, that is for instance

$$f(z) = \sum_{\alpha} \frac{\lambda_\alpha}{(1-z)^\alpha} + \frac{10!}{2311776}\frac{1}{(1-z)^2}\log\frac{1}{1-z} - \frac{26}{3}\frac{1}{1-z} \tag{2}$$

for Quicksort with 3–3 median, and by singularity analysis to compute the following expectations

Method	Mean
Quicksort	$2n\log n$
Median of 3	$(12/7)n\log n$
Median of 3–3	$1.57n\log n$

2.2. **Cumulants.** Consider a random variable of probability generating function $g(z) = \sum_n g_n z^n$. Its cumulants are defined by

$$\kappa_p(n) = \left.\frac{\partial^p}{\partial y^p} \ln g_n(e^y)\right|_{y=0}.$$

Notice that κ_1 and κ_2 respectively represent the mean and the variance of the considered distribution. Hennequin [5] showed that the cumulants of median of $2t+1$ Quicksort cost for s-ary trees are of the form

$$\kappa_p(n) = n^p K_{s,t}^p (L_{p,s,t} - (p-1)!\,\zeta(p)) + o(n^p),$$

where ζ is the zeta Riemann function and the constants $K_{s,t}$ and $L_{p,s,t}$ are rational numbers easily computed by induction.

3. Properties of the Limit Law for Internal Path Length (Binary Case)

Though the problem is still open whether there exists a close form expression of the limit law in terms of known functions, we know some properties of the limit law.

3.1. **Existence of a limit law.** Let X_n be the random variable counting the number of comparisons in an array of size n, and $Y_n = \frac{X_n - \mu_n}{n}$ the corresponding normalized random variable. Régnier showed the existence of a limit law for Y_n with almost sure convergence by using martingales.

3.2. **Method of contraction.** X_n follows the same distribution as $n-1+X_{Z_n-1}+X_{n-Z_n}$, where Z_n is uniformly drawn in the set $\{1,\cdots,n\}$: Z_n-1 and $n-Z_n$ represent the sizes of the left and right subarrays. In terms of Y_n, it rewrites into the recurrence relation

$$Y_n \stackrel{\mathcal{D}}{=} Y_{Z_n-1}\frac{Z_n-1}{n} + \overline{Y}_{n-Z_n}\frac{n-Z_n}{n} + C_n(Z_n),$$

for some computable $C_n(Z_n)$, and one can guess that the limit law of Quicksort cost is a fixed point of the equation

$$Y \stackrel{\mathcal{D}}{=} \overline{Y}\tau + \overline{\overline{Y}}(1-\tau) + C(\tau),$$

where $\overline{Y}$ and $\overline{\overline{Y}}$ are independent copies of Y, and $C(u) = 1+2u\ln u+2(1-u)\ln(1-u)$. Rösler [8, 9] established that it is true by using a method of contraction, working in a metric space of distribution, endowed with the Wasserstein metrics d_2 defined by $d_2(F,G) = \inf \|X-Y\|_2$.

Using the same method but with more precise majorizations, Fill and Janson found the following bounds on the speed of convergence: $d_2(Y_n,Y) < 2/\sqrt{n}$, and more generally $d_p(Y_n,Y) < c_p/\sqrt{n}$ for certain constants c_p. They also showed that $d_p(Y_n,Y) = O(\log n/n)$.

3.3. **Density of limit law.**

3.3.1. *Existence.* Tan and Hadjicostas [10] showed that the limit law Y of Quicksort cost admits a density, by considering the function $h_{y,z}(u) = uy + (1-u)z + C(u)$, which is clearly related to the fixed-point equation. By exchanging the axes, we get a curve with two branches r and l that are differentiable and hence admit a density.

Hence we can write

$$\mathbf{P}\big(h(U) \le t\big) = \int_{-\infty}^{t} \left(r'\,1_{[b,y+1]} - l'\,1_{[b,z+1]}\right) d\lambda = \int_{-\infty}^{t} g(y,z,t),$$

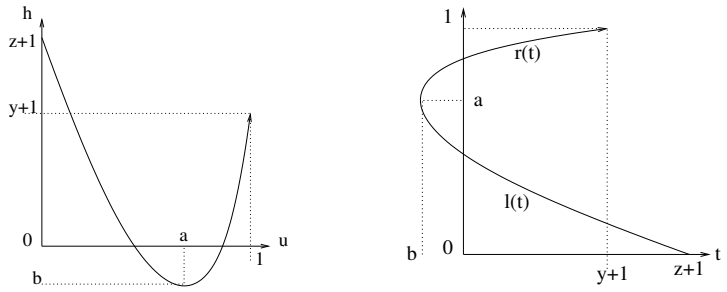

FIGURE 2. The function $h_{y,z}$ and its inverse.

(1_S is the characteristic function of the set S) and then, for all Borel set B,

$$\mu(B) = \mathbf{P}\big(UY + (1-U)\overline{Y} + C(U) \in B\big) = \int_{\mathbb{R}^2} \mathbf{P}\big(h_{Y,\overline{Y}}(U) \in N\big)\, d\mu \otimes d\nu$$
$$= \int_{\mathbb{R}^2} \int_B \left(g(y,z,s)\, d(\mu\otimes\mu)(y,z)\right) d\lambda(s) = \int_B \left(\int_{\mathbb{R}^2} g(y,z,s)\, d(\mu\otimes\mu)(y,z) \right) d\lambda(s),$$

which proves the result.

3.3.2. *Bounds on the density and its derivatives.* Fill and Janson showed that, for all integer p, there exists a constant b_p such that the characteristic function $\phi(t) = \mathbf{E}\, e^{itY}$ satisfies $|\phi(t)| \le c_p|t|^{-p}$ for all $t \in \mathbb{R}$. Using the equality

$$f^{(k)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (-it)^k e^{-itx} \phi(t)\, dt,$$

they deduce that the density f of Quicksort is C^∞ and the bounds, for all $k \in \mathbb{N}$ and $p \in \mathbb{R}^+$, $\big|f^{(k)}(x)\big| \le C_{p,k}|x|^{-p}$. In particular we have $\big|f(x)\big| \le 15.3$.

3.3.3. *Queues on limit distribution.* Knessl and Szpankowski [6] established that the left tail of the limiting distribution has a doubly exponential decay, while the right tail only has an exponential decay:

$$\begin{cases} \mathbf{P}\big(\mathcal{L}(Y_n) - \mathbf{E}\,\mathcal{L}(Y_n) \le nz\big) \sim \frac{2}{\pi} \frac{1}{\sqrt{2\log 2 - 1}} \exp\left(-\alpha \exp\left(\frac{\beta - z}{2 - \log^{-1} 2}\right)\right), \\ \mathbf{P}\big(\mathcal{L}(Y_n) - \mathbf{E}\,\mathcal{L}(Y_n) \ge ny\big) \sim a(y)\exp(-yb(y)), \end{cases}$$

where α and β are constants, and a and b are positive and polynomially bounded functions.

3.3.4. *Simulation of Quicksort Distribution.* Devroye, Fill, and Neininger devised a rejection algorithm that simulates Quicksort in a perfect way. They use a fully known function g majorizing f, and a sequence of error bounds based on the difference between the distribution function F_n of X_n and the limit distribution function F. Figure 3 shows the functions g, Quicksort density (bold curve), and successive error bounds. The algorithm stops when one goes outside an error bound, rejects if one is over the upper bound, and accepts if one is below the lesser bound.

4. m-ary Trees

The m-ary search trees are a generalization of binary search trees. We choose now up to $m-1$ medians among $m(t+1)-1$ elements, and put these medians in the same node.

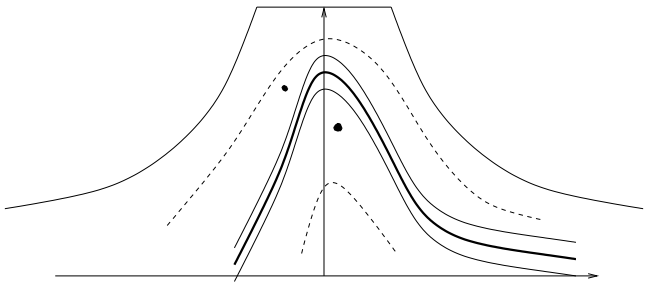

FIGURE 3. Quicksort simulation with rejection algorithm.

4.1. **Space requirement for m-ary trees.** As the number of keys occupying a node can be less than $m-1$ (it corresponds to a subarray with a size inferior to $m-1$), an issue at stake for $m>2$ is to know the number X_n of nodes (both internal and external) required to sort a given sequence of n keys.

4.1.1. *Moments.* It is possible to compute the moments of any order of the centered random variable [2]. The generating function $F(z,y)=\sum_n \mathbf{E}\left(y^{X_n}\right) z^n$ satisfies

$$D_z^{m-1}F(z,y)=(m-1)!\,F^m(z,y).$$

For the centered generating function defined by $G(z,y)=\sum_n \mathbf{E}\left(y^{X_n-\mu(n+1)}\right) z^n = y^{-\mu}F(zy^{-\mu},y)$, where μ satisfies $X_n\sim\mu n$, it translates into

$$D_z^{m-1}G(z,y)=(m-1)!\,yG^m(z,y).$$

The generating function of the kth factorial moment is $G_k(z)=D_y^kG(z,y)\big|_{y=1}$. It satisfies

$$\mathcal{L}[G_k]=k!\,(m-1)!\,(1-z)^{m-1}Q_k(z),$$

where the operator $\mathcal{L}$ is here defined by $\mathcal{L}[G]=(1-z)^{m-1}D_z^{m-1}G-m!\,G$, and Q_k is a linear combination of products of G_j's with $j<k$. The asymptotics of the variance depends on the position of the zeroes of the indicial polynom of $\mathcal{L}[G_2]$, and the limit behaviour varies with m.

4.1.2. *Gaussian limit law for $m\le 26$.* The variance is linear if $m\le 26$ because

$$G_2(z)\sim\frac{\sigma^2}{(1-z)^2}.$$

If $m\le 26$ then $\frac{X_n-\mu n}{\sigma\sqrt{n}}\to\mathcal{N}(0,1)$. Indeed, pumping moments provides an asymptotics for the G_k's

$$\begin{cases} G_{2k-1}(z)=o\left(|1-z|^{-k-1/2}\right),\\ G_{2k}(z)\sim(2k)!\,2^{-k}\sigma^{2k}(1-z)^{-k-1}, \end{cases}\qquad \text{which entails}\qquad \begin{cases} \mathbb{E}\left(\frac{X_n-\mu n}{\sigma\sqrt{n}}\right)^{2k-1}=o(1),\\ \mathbb{E}\left(\frac{X_n-\mu n}{\sigma\sqrt{n}}\right)^{2k}=\frac{(2k)!}{2^k k!}. \end{cases}$$

4.1.3. *Still an open case for $m>26$.* The variance is more than linear if $m>26$:

$$\mathbf{Var}(X_n)\sim a(n)n^{2\alpha-2}.$$

The limit law is conjectured not to be Gaussian any longer. If ever it was, then the normalization would be exotic, because for $m>26$, the limit distribution of the random variable $(X_n-\mu n)/n^{\alpha-1}$ does not exist.

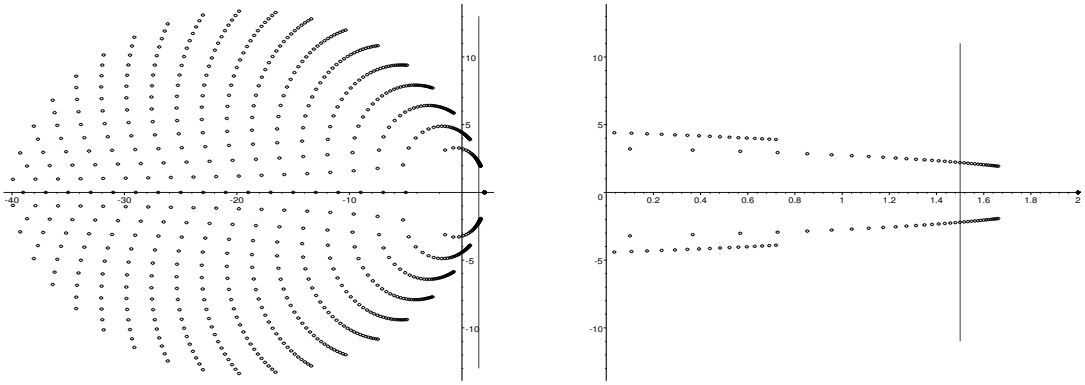

FIGURE 4. (Borrowed from [2].) Zeros of the indicial polynom $\Lambda(\vartheta)$ of $\mathcal{L}[G_1]$ for m from 5 to 40; zeros with positive real parts and the vertical line $\mathrm{Re}(\vartheta) = 3/2$ (which may be called the "phase-change line") are shown on the right.

4.2. **Profile of the tree.** Let $f_n(u)$ be the generating function of the height of leaves in m-ary trees of size n. The recurrence $f_n = um \sum_i \pi_i f_i$, where $\pi_i = \binom{n}{m-1}^{-1} \binom{n-i-1}{m-2}$ is the probability that an m-ary tree of size n has a first child of size i [3], translates into the differential equation $D_z^{m-1} F(z,u) = um!\, F(z,u)(1-z)^{1-m}$, where $F(z,u) = \sum f_n(u) z^n$. It solves to $F(z,u) \sim \lambda(u)(1-z)^{\alpha(u)}$ in the vicinity of $u = 1$. Hence $f_n(u) \sim \lambda(u)\Gamma\big(\alpha(u)\big)^{-1} \left(e^{\alpha(u)-1}\right)^{\log n}$, and according to the Quasi Powers theorem [4], the limit law of the level of the leaves in an m-ary tree is Gaussian.

It was already noticed in [4] that, heuristically, there seems to be a strong limit theorem for the profile of binary search trees. Almost sure convergence is now established for the limiting behaviour of nodes in level k of binary search trees of size n in the central region $1.2 \log n \leq k \leq 2.8 \log n$ [1], by use of martingale methods and complex analysis.

Bibliography

[1] Chauvin (Brigitte), Drmota (Michael), and Jabbour-Hattab (Jean). – The profile of binary search trees. *The Annals of Applied Probability*, vol. 11, n° 4, 2001, pp. 1042–1062.

[2] Chern (Hua-Huai) and Hwang (Hsien-Kuei). – Phase changes in random m-ary search trees and generalized quicksort. *Random Structures & Algorithms*, vol. 19, n° 3-4, 2001, pp. 316–358. – Analysis of algorithms (Krynica Morska, 2000).

[3] Durand (Marianne). – *Holonomie et applications en analyse d'algorithmes et combinatoire*. – Mémoire de DEA, Projet Algorithmes, INRIA Rocquencourt, 2000.

[4] Flajolet (Philippe) and Sedgewick (Robert). – *The average case analysis of algorithms: multivariate asymptotics and limit distributions*. – Research Report n° 3162, Institut National de Recherche en Informatique et en Automatique, 1997. 123 pages.

[5] Hennequin (Pascal). – *Analyse en moyenne d'algorithmes, tri rapide et arbres de recherche*. – Thèse de doctorat, École polytechnique, March 1991. 162 pages.

[6] Knessl (Charles) and Szpankowski (Wojciech). – Quicksort algorithm again revisited. *Discrete Mathematics & Theoretical Computer Science*, vol. 3, n° 2, 1999, pp. 43–64.

[7] Martínez (Conrado) and Roura (Salvador). – Optimal sampling strategies in quicksort and quickselect. *SIAM Journal on Computing*, vol. 31, n° 3, 2001, pp. 683–705.

[8] Rösler (U.). – On the analysis of stochastic divide and conquer algorithms. *Algorithmica*, vol. 29, n° 1-2, 2001, pp. 238–261. – Average-case analysis of algorithms (Princeton, NJ, 1998).

[9] Rösler (Uwe). – A fixed point theorem for distributions. *Stochastic Processes and their Applications*, vol. 42, n° 2, 1992, pp. 195–214.

[10] Tan (Kok Hooi) and Hadjicostas (Petros). – Some properties of a limiting distribution in Quicksort. *Statistics & Probability Letters*, vol. 25, n° 1, 1995, pp. 87–94.