Partial Quicksort and Quickpartitionsort

Conrado Martínez, Uwe Rösler

Partial Quicksort and Quickpartitionsort

Conrado Martínez and Uwe Rösler

Dept. Llenguatges i Sistemes Informatics, Univ. Politecnica de Catalunya, Spain
Mathematisches Seminar, Christian-Albrechts-Universität zu Kiel, Germany

Partial Quicksort sorts the l smallest elements in a list of length n. We provide a complete running time analysis for this combination of Find and Quicksort. Further we give some optimal adapted versions, called Partition Quicksort, with an asymptotic running time $c_1 l \ln l + c_2 l + n + o(n)$. The constant c_1 can be as small as the information theoretic lower bound $\log_2 e$.

Keywords: searching, sorting, Find, Quicksort, divide and conquer algorithm, random algorithm, running time analysis, asymptotic distribution, optimal adapted algorithms, stochastic fixed point equation, contraction method

1 Introduction

Partial Quicksort sorts the l-th smallest elements of a list. The basic idea is a combination of Quickselect or Find, introduced by Hoare [4], finding the l-th smallest and of Quicksort, also invented by Hoare [5], for quick sorting. Basically do first Find to find the l-th smallest and then quicksort all l smallest using all available information obtained so far (splitting). (For the running time analysis it is more convenient, to quicksort on the run, whenever possible.)

Both stochastic divide-and-conquer algorithms Find [3] [2] [6] and Quicksort [10] [11] [14] are well studied. Many of the mathematical tools developed there [12] apply and allow a complete analysis of Partial Quicksort and also finding the optimal adapted algorithms Quickpartitionsort. We will point out the basic procedure for Partial Quicksort.

For simplicity we discuss first standard Find for finding the l-th and then standard Quicksort for sorting. In more detail: Choose some pivot element from the list, usually by random with a uniform distribution, and split the list S into the strictly smaller ones $S_<$ and the strictly larger ones $S_>$. If the l-th is in $S_<$ then recall recursively the algorithm for $S_<$. If the l-th is in $S_>$ (or the pivot) then sort $S_<$ and recall recursively the algorithm for $S_>$. If neither, then recall the algorithm for $S_<$. (A pseudo-code description of this recursion is given in [7].)

The recursive formula for the rv $X(S,l)$ of comparisons in order to find the sorted l smallest out of the set S of different numbers is

$$\begin{aligned} X(S,l) &= |S| - 1 + \mathbb{1}_{|S_<|<l-1}(X_1(S_<,|S_<|) + X_2(S_>, l-|S_<|-1)) \\ &+ \mathbb{1}_{|S_<|=l-1} X_1(S_<,|S_<|) + \mathbb{1}_{|S_<|\geq l} X_1(S_<,l). \end{aligned} \tag{1}$$

The rvs X_1, X_2 denote the number of comparisons (left and right branch) using recursively the same algorithm. The rv $I = |S_<| + 1$ is independent of the X_i-rvs and given $S_<$ and $S_>$ the rvs X_1, X_2 are

independent. Notice, the distribution of $X(S,|S|)$ is the Quicksort distribution sorting all numbers by some specified Quicksort version. The rv $I = I(n,l)$ denotes the rank of the pivot after comparisons and has values on $\{1,2,\ldots,n\}$.

The distribution of $X(S,l)$ depends only on $|S|$ and l. We use $X(S,l) \stackrel{\mathcal{D}}{=} X(|S|,l)$. This is due to the internal randomness and is true for any input S. We skip the induction on $n = |S|$ via equation (1).

The equation (1) determines recursively the distribution of comparisons $X(n,l)$ in order to find the sorted l smallest out of n different numbers,

$$\begin{aligned} X(n,l) &\stackrel{\mathcal{D}}{=} n-1+\mathbb{1}_{I<l}(X_1(I-1,I-1)+X_2(n-I,l-I)) \\ &+ \mathbb{1}_{I=l}X_1(I-1,I-1)+\mathbb{1}_{I>l}X_1(I-1,l). \end{aligned} \tag{2}$$

The rvs $I = I(n,l), X_i(j,k),\ i,j,k \in \mathbb{N}, i = 1,2,\ k \le j < n$ are independent. The rv I has values in $\{1,2,\ldots,n\}$, $X_i(j,\cdot)$ have the same distribution as $X(j,\cdot)$. The equation (2) is a consequence of (1).

From Equation 2 we obtain the best and worst performance of the algorithm. Choosing a pivot by random with a uniform distribution, the best is by picking incidentally the l-th largest for the Find procedure and then doing Quicksort in its best. For standard Quicksort this is picking incidentally the pivot as an median. We face the worst behavior, if our pivot is the largest all the time.

From equation (2) we obtain for the expectation $a(n,l) = EX(n,l)$

$$\begin{aligned} a(n,l) &= n-1+\sum_{j=1}^{l-1} P(I=j)(a(j-1,j-1)+a(n-j,l-j)) \\ &+ P(I=l)a(l-1,l-1)+\sum_{j=l+1}^{n} P(I=j)a(j-1,l). \end{aligned}$$

The term $a(j,j)$ is the expectation of sorting j numbers by Quicksort. For a uniformly distributed rank of the pivot we have an explicit solution [7]

$$a(n,l) = 2n+2(n+1)H_n - 2(n+3-l)H_{n+1-l} - 6l + 6.$$

H_n denotes the n-th harmonic number

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + \gamma + o(1)$$

γ the Euler constant.

Now to the distribution. For simplicity we stay with a uniform distributed rank of the pivot. The correct normalization for the $X(n,\cdot)$ rv is

$$Y^n(\frac{l}{n}) = \frac{X(n,l)-a(n,l)}{n}.$$

We obtain the recursion, $I = I(n,l)$

$$Y^n(\frac{l}{n}) \stackrel{\mathcal{D}}{=} \mathbb{1}_{I<l}\frac{n-I}{n}Y_2^{n-I}(\frac{l-I}{n-I}) + \mathbb{1}_{I>l}\frac{I-1}{n}Y_1^{I-1}(\frac{l}{I-1}) + C^n(\frac{l}{n}) \tag{3}$$

where

$$\begin{aligned} C^n(\frac{l}{n}) &= \mathbb{1}_{I\le l}\frac{I-1}{n}Y_1^{I-1}(1)+\frac{n-1}{n}-\frac{a(n,l)}{n} \\ &+ \mathbb{1}_{I\le l}\frac{a(I-1,I-1)}{n}+\mathbb{1}_{I<l}\frac{a(n-I,l-I)}{n}+\mathbb{1}_{I>l}\frac{a(I-1,l)}{n} \end{aligned} \tag{4}$$

The distribution of $Y^j(1)$ is the Quicksort distribution to sort j elements. Since we consider only distributions, we do not change the distribution of $Y^n(\cdot)$ if we replace $Y_1^j(1)$ by other rvs $Q(j)$, independent of everything so far, with the (same) j-Quicksort distribution. We use this replacement in the sequel.

The equation (3) allows an analysis of the asymptotic of $Y^n(\frac{l}{n})$ in n. We shall give a version of process valued rvs Y^n with the same one dimensional marginals. The definition of the processes Y^n is via

$$(Y^n(\frac{l}{n}))_l \stackrel{\mathcal{D}}{=} (\mathbb{1}_{I<l}\frac{n-I}{n}Y_2^{n-I}(\frac{l-I}{n-I})+\mathbb{1}_{I>l}\frac{I-1}{n}Y_1^{I-1}(\frac{l}{I-1})+C^n(\frac{l}{n}))_l \tag{5}$$

Extend $Y^n(\cdot)$ nicely (continuous linear) to a rv Y^n with values in function space D [1], the space of cadlag functions on the unit interval. The process Y^n will converge (see [6]) to a process Y satisfying the stochastic fixed point equation

$$(Y(t))_{t\in[0,1]} \stackrel{\mathcal{D}}{=} (\mathbb{1}_{U\le t}(1-U)Y_2(\frac{t-U}{1-U})+(\mathbb{1}_{U>t}UY_1(\frac{t}{U})+C(t))_{t\in[0,1]} \tag{6}$$

on D. The rvs Y_1, Y_2, U, Q are independent, Y_1 and Y_2 have the distribution μ, U is uniformly distributed on the unit interval I and Q has a limiting Quicksort distribution. The cost function C is given by

$$\begin{aligned} C(t) &= 2U\ln U+\mathbb{1}_{U>t}(2U-1+2(1-t)\ln(1-t)-2(U-t)\ln(U-t)) \\ &\quad +\mathbb{1}_{U\le t}(1+2(1-U)\ln(1-U)+UQ). \end{aligned}$$

Let K be the operator on the set $M(D)$ of probability measures on D to itself defined by

$$K(\mu)=\mathfrak{L}((\mathbb{1}_{U\le t}(1-U)Y_2(\frac{t-U}{1-U})+\mathbb{1}_{U>t}UY_1(\frac{t}{U})+C(t))_{t\in[0,1]})$$

Here $Y_1, Y_2, (U, C)$ are independent rvs, Y_1, Y_2 have distribution μ and (U, C) are as above. The symbol $\mathfrak{L}$ denotes the distribution of a random variable.

The existence of Y and convergence of Y^n to Y is given by the following theorem, which is provided in [6] in a more general form. The metric d on processes is the uniform Wasserstein metric,

$$d(\mu,\nu)=\inf E\sup_t|X(t)-Y(t)|$$

The infimum is taken over all processes (X, Y) with marginal distributions ν and μ.

Theorem 1. *The sequence $K^n(\delta_0)$ converges to a fixed point $\nu=\mathfrak{L}(Y)$ of K with respect to the uniform Wassserstein exponentially fast. The process Y^n (5) converges to the fixed point Y of K in terms of finite dimensional distributions. All one-dimensional distributions $Y^n(t),\ n\in\mathbb{N}$ have finite exponential moments of all order and converge to those of Y.*

The K^n convergence is much stronger (uniformly) than the Y^n convergence. The trouble are pathwise jumps of the process. The example 2-Find [3] provides convergence in Skorodhod topology for nice versions of the processes. However the 3-version converges in (finite dimensional) distribution, but not in Skorodhod topology. The basic trouble is a process like $Z_n = \mathbb{1}_{[.5-1/n,.5)}$ converging to $Z \equiv 0$ pointwise in t and in the weak sense, convergence of finite dimensional distributions. However no version of Z_n will converges in Skorodhod topology or uniformly.

In Knof-Roesler [6] the recursive setting is more general in D,

$$Y^n \stackrel{\mathcal{D}}{=} \sum_{i\in\mathbb{N}} A_i^n Y_i^{I_i^n} \circ B_i^n + C^n \tag{7}$$

Here $((A_i^n, B_i^n, I_i^n)_i, C^n), Y_k^j,\ k,j \in \mathbb{N}$ are independent. A_i^n, C^n, B_i^n have values in D, B_i^n is a piecewise increasing random time change. Under certain assumptions Y^n converges in distribution to Y satisfying

$$Y \stackrel{\mathcal{D}}{=} \sum_{i\in\mathbb{N}} A_i Y_i \circ B_i + C.$$

All one dimensional distributions $Y^n(t)$ have finite exponential moments of all orders and converge to those of Y. (All details are given in more generality in [6] and are skipped here.)

Dealing with the positive case (everything positive) is easier, since the distribution of Y^n converges monotone in stochastic order to a limit. (And for suitable versions also point wise.) Otherwise, loosing the monotonicity, we have to circumvent this difficulty by showing certain sums converge absolutely [6]. Partial Quicksort is a nice example, since C takes positive and negative values in a natural, non trivial way and still the theory applies and the assumptions are satisfied. The running time analysis of standard Partial Quicksort is complete.

Analogous, but with more technical effort, we can analyze different versions of Find and Quicksort, choosing the pivot differently, e.g. as k-median. Find uses in the optimal case [8] asymptotically $n + \inf\{l, n-l\} + o(n)$ comparisons in order to find the l-th out of n. Afterwards Quicksort uses $Q(l) = (Q(l) - a(l,l)) + a(l,l)$ comparisons. The dominant term is the deterministic term $a(l,l)$ of the order $cl \ln l$. The constant c varies with the Quicksort version. c has a lower bound $\ln_2 e$. We can do slightly better, even asymptotically in the second leading term, performing approximate Find and using the best Quicksort version. That is what adaptive Quickpartitionsort does.

Quickpartitionsort chooses first a pivot k which rank is slightly larger than l with high probability. Then quicksort the k smallest. In our analysis we use a crude version, take a rv k_1 from a sample drawn. If $k_1 \geq l$ take $k = k_1$ and otherwise repeat the procedure. For the repetition we may forget the k_1 and start afresh, since asymptotically the event of repetition is negligible in the analysis. The overshoot $I > l$ is also asymptotically negligible, since we will show $a(n,l) - a(n,I)$ is of order $o(n)$.

Quickpartitionsort is kind of optimal in this sense of the second leading term. We can not prove rigorous mathematical optimality, since the lack of a (tight) lower bound besides $n - l$ plus best Quicksort(l). In the remaining parts of the paper we state and proof this statement.

2 Quickpartitionsort

Quickpartitionsort is an Algorithm with input a set of n different reals and output the l-th smallest numbers as a sorted list. The algorithm Quickpartitionsort $QP(m_n, \epsilon_n, Q_n),\ n \in \mathbb{N}$ depends on three parameters,

the parameter $1 \leq m_n \leq n$ for the sample size, the ϵ_n determines the choice of the pivot and Q_n is a Quicksort variant. n is the size of the input.

Quickpartitionsort consists of two basic steps, first a partitioning and then a quick sorting. In more detail, for given $n \in \mathbb{N}$

- For large l, more precisely if $\frac{m_n}{m_n+1} - \frac{l}{n} < \epsilon_n$, continue with quick sorting using Q_n.
- If $\frac{m_n}{m_n+1} - \frac{l}{n} \geq \epsilon_n$ define $k_n(l)$ as the $k \in \{1, 2, \ldots, n\}$ with $\frac{k}{m_n+1} - \frac{l}{n}$ is strictly positive and as close as possible to ϵ_n. Take m_n samples $U_1, U_2, \ldots, U_{m_n}$ without replacement and with uniform distribution on the list objects. Order them into

  $$U_{1:m_m} < U_{2:m_n} < \ldots < U_{m_n:m_n}$$

  and take as pivot U the $k_n(l)$-th ordered one $U_{k_n(l):m_n}$ in the sample.
- Compare any element of the list with the pivot element and form the list of numbers strictly smaller or equal to the list. Let $I_n(l) = I(n, l)$ be the size of this set (=rank of U in the whole set).
- If $I_n(l)$ is strictly smaller than l, then start the algorithm afresh. If not continue with quick sorting that list.
- After quick sorting the l-th smallest of the ordered partial list are the output.

Clearly this algorithms terminates a.e. and does the job.

Before we give results, let us introduce some notation on order statistics and give some well known statements. Let $U_1, U_2, \ldots U_n$ be iid rvs with a continuous distribution function F. The empirical distribution function F is

$$F_n(x) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{U_i \leq x}.$$

The empirical distribution function converges pointwise to the underlying distribution function F. The expectation $EF_n(x)$ is $F(x)$ and the variance $\operatorname{var} F_n(x) = \frac{F(x)(1-F(x))}{n}$. The rank of U_i within $U_1, \ldots, U_n$ is $nF_n(U_i)$. The k-th smallest within $U_1, \ldots, U_n$ is denoted by $U_{k:n}$. Notice $k = nF_n(U_{k:n})$ almost everywhere. If the U are uniformly distributed, then the expectation of the k-th smallest is $EU_{k:n} = \frac{k}{n+1}$ and the variance $\operatorname{var}(U_{k:n}) = \frac{k(k+1)}{(n+1)^2(n+2)} \leq \frac{1}{n}$.

Theorem 2. *For every $n \in \mathbb{N}$ fix a Quicksort version Q_n with internal randomness, such that the expected average sorting time for n objects is of the order $c_1 n \ln n + c_2 n + o(n)$ for some constants $c_1 > 0$, $c_2 \in \mathbb{R}$. Assume*

$$m_n \to_{n\to\infty} \infty,\ 0 < \epsilon_n \to_n 0,\ m_n \epsilon_n^2 \to_n \infty,\ \frac{m_n}{n\epsilon_n} \to_n 0.$$

Then Quickpartitionsort $QP(m_n, \epsilon_n, Q_n)$ uses at most in expectation asymptotically (in n)

$$c_1 l \ln l + n + c_2 l + o(n)$$

comparisons uniformly in l for an input of size n.

Proof: Let S be the input of size n and $X(S,l)$ be the number of comparisons during the performance of Quickpartitionsort $QP(m_n, \epsilon_n, Q_n)$. Since we have a stochastic divide-and-conquer algorithms, we obtain recursive equations of the X-rvs. The X-rvs depend on the external input. By the internal randomness of the algorithm, which is independent of the input, we can show by induction that the distribution of the rvs X depend on the size of the input, but not on the actual input. The induction is tedious, but easy and we skip it. As a result we can talk of finding the ordered l-th smallest for a list of size n, without specifying the input list. Therefore we are allowed to use $X(n,l)$ and the distribution of this rv is the same for every input of size n.

Case: Immediate quicksorting, $\frac{m_n}{m_n+1} - \frac{l}{n} < \epsilon_n$.
Then the total number of comparisons is by the Quicksort step,

$$\begin{aligned} c_1 n \ln n &+ c_2 n - c_1 l \ln l - c_2 l + o(n) \\ &= nc_1 \ln \frac{l}{n} - n\left(\frac{1}{m_n + 1 + \epsilon_n}\right) \ln \frac{l}{n} + nc_2\left(1 - \frac{l}{n}\right) + o(n) = o(n) \end{aligned}$$

Case: A partitioning step is done, $\frac{m_n}{m_n+1} - \frac{l}{n} \geq \epsilon_n$.

Without loss of generality we assume the input is a sequence $U_1, U_2, \ldots, U_n$ of iid rvs with a uniform distribution, independent of any internal randomness. (Since the input does not matter, we can take n iid rvs independent of the algorithm. The continuous distribution ensures a.e. n different reals.) Without loss of generality let $U_1, \ldots, U_{m_n}$ be the drawn random sample of size m_n. (Any other random sample would do the same job, since in the following only the distribution of the rank $I_n(l)$ matters.) Use any reasonable algorithm to sort the m_n-sample, for example the fixed Quicksort version. An upper bound for the necessary comparisons is $c_1 m_n \ln m_n + c_2 m_n + o(m_n)$ and for a lower $n-1$. (Any reasonable algorithm will do the job, for definiteness of the Quickpartitionsort one should fix some version.)

Let $\frac{k_n(l)}{m_n+1} - \frac{l}{n} := \epsilon_n(l) > 0$ be given as described above. Let $U = U_{k_n(l):m_n}$ be the k-th order statistic within $U_1, \ldots, U_{m_n}$. Let $I_n(l)$ be the rank of U within the sequence $U_1, \ldots, U_n$. Recall $EU = \frac{k_n(l)}{m_n+1}$ and the variance $\operatorname{var}(U) = \frac{k_n(l)(k_n(l)+1)}{(m_n+1)^2(m_n+2)} \leq \frac{1}{m_n}$.

An easy calculation shows

$$EI_n(l) = \frac{m_n + 1}{2} + (n - m_n)EU$$

and the variance is

$$\operatorname{var}(I_n(l)) = (n - m_n)^2 \operatorname{var}(U) + (n - m_n)EU(1 - U) \leq \frac{n^2}{m_n} + n.$$

Notice

$$E\frac{I_n(l)}{n} - \frac{l}{n} = \frac{n - m_n}{n}\epsilon_n(l) + \frac{m_n + 1}{2n} - \frac{m_n l}{n^2} = \epsilon_n(1 + o(1)) > 0$$

for n sufficiently large. Therefore for n sufficiently large, for simplicity $I = I_n(l)$

$$P(I < l) \leq P(I - EI < l - EI) \leq \frac{\operatorname{var}(I)}{(EI - l)^2} \leq \frac{o(1)}{m_n \epsilon_n^2} \to_n 0. \tag{8}$$

This estimate is uniform in l of consideration.

Choose some δ_n satisfying $\liminf_n \frac{\delta_n}{\epsilon_n} > 1$ and $\frac{\ln n}{m_n \delta_n^2} \to_n 0$. This is always possible, e.g. $\delta_n = \epsilon_n \sqrt{\ln n}$.

For later use we need, n large,

$$\begin{aligned}
P(I > l + \delta_n n) &\leq P(I - EI > l - EI + \delta_n n) \leq \frac{\operatorname{var} I}{(-EI) + l + \delta_n n)^2} \\
&\leq \frac{o(1)}{m_n \epsilon_n^2 (\frac{\delta_n}{\epsilon_n} + o(1))^2} = O(\frac{1}{m_n \epsilon_n^2}) = o(\frac{1}{\ln n}) \\
\frac{1}{n} E(I \mid l \leq I) &\leq \frac{1}{n} E(\mathbb{1}_{l \leq I} I)(1 + P(l > I)) \leq \delta_n (1 + o(1)) \\
\frac{1}{n} E(I \ln I \mid l \leq I) &\leq \frac{1}{n} E(\mathbb{1}_{l \leq I} I)(1 + P(l > I)) \\
&\leq (1 + o(1))(\frac{1}{n} E(\mathbb{1}_{l \leq I \leq l + \delta_n n} I \ln I) \\
&\quad + \frac{1}{n} E(\mathbb{1}_{l + \delta_n n < I} I \ln I)) \\
&\leq (1 + o(1))((\frac{l}{n} + \delta_n) \ln((l + \delta_n) n) \\
&+ (\frac{l}{n} + \delta_n) \ln n P(l + \delta_n n < I) \leq \frac{l}{n} \ln l + o(1)
\end{aligned}$$

For the last estimate the crucial point is $|\frac{l + \delta_n n}{n} \ln(l + \delta_n n) - \frac{l}{n} \ln l| = o(1)$. For that show the function $x \mapsto \ln(1 + \frac{b}{x})$ is monotone increasing (look at the second derivative $-\frac{b^2}{(b+x)x(x+b)}$) for positive b.

Combining these shows

$$E(\frac{X(n,l)}{n} \mid l \leq I) \leq c_1 l \ln l + c_2 l + n + o(l)$$

uniformly in l.

Let $Y_j, j \in \mathbb{N}$ be iid Bernoulli rv with parameter $p = p = P(l \leq I_n(l))$. Let τ be the smallest j with $Y_j = 1$. We interpretate $Y_j = 1$ as the i-th try of $I_n(l)$ is greater equal to l. Then we obtain finally

$$EX_n(l) \leq m_n^2 E\tau + E(X(n,l) \mid l \leq I_n(l)) = c_1 l \ln l + c_2 l + n + o(n).$$

($E\tau = \frac{1}{p}$.) This estimation is uniformly in l. q.e.d.

By an example we show that there exist m_n, ϵ_n satisfying the requirements of the theorem. Take $m_n = \ln^5 n,\ \epsilon_n = \frac{1}{\ln^2 n},\ \delta_n = \frac{1}{\ln n}$.

Natural candidates for Quicksort versions are the k-median versions and also the asymptotically optimal Quicksort variant. For the k-median the asymptotic expectation of comparisons is $c_{1,k} n \ln n + c_{2,k} n + o(n)$, [13]. The constant $c_{1,k}$ decrease in k to the optimal value $c_{1,\infty} = \log_2 e$. (Optimal by the information theoretic lower bound.) For the optimal Quicksort [9] exists a constant c_2 such that the expectation is asymptotically bounded by $c_{1,\infty} n \ln n + c_2 n + o(n)$. We can apply Theorem 2 and obtain a leading term of the order $c_{1,\infty} l \ln l$ and linear terms $n + c_2 l$ with unknown c_2.

References

[1] Patrick Billingsley. *Convergence of Probability Measures*. Wiley Series in Probability and Statistics, 1968.

[2] R. Grübel. On the median-of-k version of Hoare's selection algorithm. *Theoretical Informatics and Applications*, 33(2):177–192, 1999.

[3] Rudolf Grübel and Uwe Rösler. Asymptotic distribution theory for Hoare's selection algorithm. *Advances in Applied Probability*, 28:252–269, 1996.

[4] Charles A.R. Hoare. PARTITION (Algorithm 63);QUICKSORT (Algorithm 64);FIND (Algorithm 65). *Communication of the Association for Computing Machinery*, 4:321–322, 1961.

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

[6] Diether Knof and Uwe Roesler. The analysis of find or perpetuities on cadlag functions. *Discrete Mathematics and Theoretical Computer Science*. Accepted.

[7] Conrado Martínez. Partial quicksort. *Alcom-FT Technical Report Series*, ALCOM-FT-RR-03-50, 2004.

[8] Conrado Martínez, Daniel Panario, and Alfred Viola. Adaptive sampling strategies for quickselect. *ACM Transactions on Algorithms*, 2008. accepted.

[9] Conrado Martínez and Salvador Roura. Optimal sampling strategies in quicksort and quickselect. *SIAM Journal Computing*, 31(3):683–705, 2001.

[10] Uwe Roesler. A limit theorem for quicksort. *Theoretical Informatics and Applications*, 25:85–100, 1991.

[11] Uwe Roesler. A fixed point theorem for distributions. *Stochastic Processes and their Applications*, 42:195–214, 1992.

[12] Uwe Roesler. *Dynamics of complex and irregular systems, Bielefeld Encount. Math. Phys. VIII, Bielefeld 1991*, chapter The weighted branching process., pages 154–165. World Sci. Publishing, River Edge, NJ, 1993.

[13] Uwe Roesler. On the analysis of stochastic divide and conquer algorithms. *Algorithmica*, 29:238–261, 2001.

[14] Uwe Roesler and Ludger Rüschendorf. The contraction method for recursive algorithms. *Algorithmica*, 29:3–33, 2001.