Electronic Colloquium on Computational Complexity, Report No. 59 (2004)

Randomized Quicksort and the Entropy of the Random Number Generator

Beatrice List, Markus Maucher, Uwe Schöning and Rainer Schuler

Abt. Theoretische Informatik, Universität Ulm, 89069 Ulm, Germany

June 21, 2004

Abstract

The worst-case complexity of an implementation of Quicksort depends on the random number generator that is used to select the pivot elements. In this paper we estimate the expected number of comparisons of Quicksort as a function in the entropy of the random source. We give upper and lower bounds and show that the expected number of comparisons increases from $n \log n$ to n^2, if the entropy of the random source is bounded. As examples we show explicit bounds for distributions with bounded min-entropy, the geometrical distribution and the δ-random source.

1 Introduction

Randomized QuickSort is the well known version of QuickSort (invented by Hoare [Ho]) where the array element for splitting the array in two parts (the ”pivot” element) is selected at random. It is also well known that the expected number of comparisons (for *every* input permutation of the array elements) is $(2 \ln 2) \cdot n \log_2 n - \Theta(n)$. Here, the expectation is taken over the random choices done in the algorithm. This analysis assumes random numbers which are independent and uniformly distributed.

Here we analyze randomized QuickSort without assuming such an ”high entropy” of the underlying random source. Using a random number generator with a low entropy can result in a worst-case behavior that can go up to $\Theta(n^2)$. An extreme example is a ”very bad” random number generator that produces only ”1” as output. That is, in each recursive call of QuickSort the first array element is selected as pivot element. A worst case input in this case is the already sorted array.

Related work has been done by Karloff and Raghavan [KR] (see also [To]) where the special case of a linear congruence generator is considered and a worst-case behavior of $\Omega(n^2)$ is shown.

Recursion for expected number of comparisons

Let $T_\pi(n)$ be the expected number of comparisons done by randomized QuickSort, when operating on an input array $(a[1], \ldots, a[n])$ whose elements are permuted according to $\pi \in S_n$, that is,

$$a[\pi(1)] < a[\pi(2)] < \cdots < a[\pi(n)],$$

where S_n is the set of all permutations on $\{1, \ldots, n\}$.

Let X be a random variable taking values between 1 and n (not necessarily under uniform distribution) which models the random number generator that is used to pick out a pivot element $a[X]$.

ISSN 1433-8092

We obtain the following recursion for the expected complexity (i.e. number of comparisons) $T(n) = \max_{\pi \in S_n} T_\pi(n)$. We have $T(n) = 0$ for $n \leq 1$; and for $n > 1$ we get

$$\begin{aligned} T(n) &= \max_{\pi \in S_n} T_\pi(n) \\ &= (n-1) + \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot (T_\pi(i-1) + T_\pi(n-i)) \\ &\leq (n-1) + \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot \left(\max_{\Phi \in S_{i-1}} T_\Phi(i-1) + \max_{\Psi \in S_{n-i}} T_\Psi(n-i) \right) \\ &= (n-1) + \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot (T(i-1) + T(n-i)) \end{aligned}$$

That is, there are $n-1$ comparisons with the selected pivot element, and depending on the rank i of the pivot element within the array, there are $T(i-1)$ and $T(n-i)$ additional comparisons. Here p_i is the probability that the pivot element has rank i within the ordering of the array, that is, $p_i = Pr(\pi(X) = i)$. If the rank is not uniformly distributed among the numbers 1 to n, a worst case input permutation can be constructed such that the middle ranks receive relatively low probability and the extreme ranks (close to 1 or close to n) get relatively high probability, resulting in a large expected number of comparisons.

We give upper and lower bounds on the expected number $T(n)$ of comparisons. Lower bounds are given with respect to a fixed input sequence (the already sorted list of elements).

We can show (see Theorem 1) that $T(n) \leq g(n) \cdot n \cdot \log_2 n$ for any function $g(n)$ greater than $1/\left(\min_\pi \sum_{i=1}^{n} p_i H\left(i/n\right)\right)$, where $H\left(i/n\right)$ is the binary entropy function. Note that $\min_\pi \sum_{i=1}^{n} p_i H\left(i/n\right)$ is independent of the permutation of the elements, i.e. is identical for all distributions p and q such that $p_i = q_{\pi(i)}$ for all i and some permutation π.

The lower bound (see Theorem 3 and 4) is derived for a fixed permutation (the sorted list of elements), where we can assume that the order is preserved in all recursive calls of QuickSort. Therefore the lower bound $T(n) \geq n \cdot g(n)$ (Theorem 4) is w.r.t. any function $g(n)$ less than $1/\sum_{i=1}^{n} p_i H\left(i/(n+1)\right)$, where p_i is the probability of selecting $a[i]$ as a pivot element.

2 Upper bound on the number of expected comparisons

Let (P_n) denote a sequence of probability distributions where $P_n = (p_{1,n}, \ldots, p_{n,n})$ is a distribution on $(1, \ldots, n)$. In the following we use p_i to denote $p_{i,n}$, since n is determined by the size of the array.

Theorem 1 *We have $T(n) \leq g(n) n \log_2 n$ for any monotone increasing function g with the property*

$$g(n) \geq \left(\min_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right) \right)^{-1}$$

where $H(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function (Shannon entropy).

Proof. Using the above recursion for $T(n)$ we obtain

$$\begin{aligned}
T(n) &= (n-1) + \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot (T(i-1) + T(n-i)) \\
&\leq n + \max_{\pi} \sum_{i=1}^{n} p_i \cdot (g(i-1)(i-1)\log_2(i-1) + g(n-i)(n-i)\log_2(n-i)) \\
&\leq n + g(n)n \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot \left(\frac{i}{n}\log_2 i + \left(1-\frac{i}{n}\right)\log_2(n-i)\right) \\
&= n + g(n)n \max_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot \left(\frac{i}{n}\log_2 \frac{i}{n} + \left(1-\frac{i}{n}\right)\log_2\left(1-\frac{i}{n}\right) + \log_2 n\right) \\
&= n + g(n)n\log_2 n - g(n)n \min_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right)
\end{aligned}$$

To finish the induction proof, this last expression should be at most $g(n)n\log_2 n$. This holds if and only if

$$g(n) \;\geq\; \left(\min_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right)\right)^{-1}$$

as claimed. □

Example: In the standard case of a uniform distribution $p_i = \frac{1}{n}$ we obtain:

$$g(n) \;\geq\; \left(\frac{1}{n} \cdot \sum_{i=1}^{n} H\left(\frac{i}{n}\right)\right)^{-1} .$$

This is asymptotically equal to

$$\left(\int_0^1 H(x)dx\right)^{-1} \;=\; 2\ln 2 \;\approx\; 1.38 \;\; .$$

Another Example: In the median-of-3 version of QuickSort (cf. [Kn,SF]), 3 different elements are picked uniformly at random and the median of the 3 is used as the pivot element. In this case $p_i = \frac{6(i-1)(n-i)}{n(n-1)(n-2)}$. Here the constant factor of the $n\log n$-term can be asymptotically estimated by

$$\left(6\int_0^1 x(1-x)H(x)dx\right)^{-1} \;=\; \frac{12\ln 2}{7} \;\approx\; 1.18$$

We ignore here the additional number of comparisons between the 3 elements to find out their median – but this does not have an influence asymptotically.

Sorting the probabilities

Using the symmetry of the function H around $\frac{1}{2}$ and its monotonicity, we get:

$$\min_{\pi \in S_n} \sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right) \;\geq\; \min_{\pi \in S_n} \sum_{j=0}^{n-1} q_j \cdot H\left(\frac{j}{2n}\right) \;\; .$$

Here, the q_j are a reordering of the p_i in the following way (assuming n is even):

$$\begin{array}{rclrcl} q_0 & = & p_n & q_1 & = & p_1 \\ q_2 & = & p_{n-1} & q_3 & = & p_2 \\ & \vdots & & & \vdots & \\ q_{n-2} & = & p_{n/2} & q_{n-1} & = & p_{n/2-1} \end{array}$$

This new representation has the advantage that the H-values in the sum are in increasing order, and we can determine which permutation $\pi \in S_n$ actually achieves the minimum. Namely, the minimum is achieved if the q_j are ordered in decreasing order. (This is in accordance with the statement in the introduction that the worst case is associated with the situation that the extreme ranks occur with higher probability than the middle ranks.)

Lemma 2 *Given a sum of the following form*

$$\sum_{j=1}^{n} a_j b_{\pi(j)}, \quad a_j, b_j \geq 0$$

where the a_j are sorted in strictly increasing order and the permutation π can be chosen arbitrarily, the minimum value of the sum occurs when the permutation π is such that the $b_{\pi(j)}$ are sorted in decreasing order.

Proof. Suppose that two elements b, b' are in the ”wrong” order, i.e. $b < b'$. We compare the situation before and after exchanging b and b':

$$(a_i b + a_j b') - (a_i b' + a_j b) \;=\; (a_i - a_j)(b - b') \;<\; 0$$

This means, interchanging b and b' this way strictly decreases the value of the sum. Furthermore, it is easy to see that the decreasingly sorted order can always be achieved by swapping two elements which are in the ”wrong” order (e.g. like in the BubbleSort algorithm). □

3 A lower bound

As we saw in Section 1, the running time of QuickSort is given by the recursion

$$T(n) = n - 1 + \sum_{i=1}^{n} p_i \cdot (T(i-1) + T(n-i)),$$

where p_i is the probability of choosing the element with rank i as pivot element.

To estimate a lower bound for the worst-case running time of QuickSort, we consider as input the already sorted array of numbers. Further we assume that the partitioning step of QuickSort leaves the elements of the two sub-arrays in the same relative order as in the input array.

Recall that pivot-elements are chosen according to a sequence of probability distributions (P_i), where distribution P_i defines the probabilities on arrays of size i, i.e. $P_i = (p_{i,1}, \ldots, p_{i,i})$. Note that if the $p_{i,j}$ are sorted in decreasing order, then a worst-case input is the already sorted sequence of numbers. In fact, if the sequence of probability distributions (P_i) is sufficiently *uniform*, it should be possible to construct a worst-case input by sorting probabilities as described in Section 2.

Theorem 3 *For any sequence of probability distributions* (P_n) *it holds that* $T(n) \geq c \cdot g(n) \cdot n$*, for some constant* $c > 0$*, if for all* $n > n_0$*,* g *satisfies the two conditions*

$$g(n) \leq \left(\frac{n}{n-1} \cdot \sum_{i=1}^{n} p_{i,n} \left(1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2} \right) \right)^{-1}$$

and

$$\frac{g(i)}{g(n)} \geq \frac{i}{n} \quad \textit{for all} \quad 0 \leq i \leq n.$$

Proof. Let $P = (p_1, \ldots, p_n)$ be a distribution where p_i is the probability that we choose as a pivot element the element with rank i. For $n > 2$, it holds

$$\begin{aligned} T(i-1) + T(n-i) &\geq (i-1) \cdot g(i-1) + (n-i) \cdot g(n-i) \\ &= n \cdot g(n) \cdot \left(\frac{(i-1) \cdot g(i-1)}{n \cdot g(n)} + \frac{(n-i) \cdot g(n-i)}{n \cdot g(n)} \right) \\ &\geq n \cdot g(n) \cdot \left(\frac{(i-1)^2}{n^2} + \frac{(n-i)^2}{n^2} \right) \\ &= n \cdot g(n) - n \cdot g(n) \cdot \left(1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2} \right). \end{aligned}$$

Therefore,

$$\begin{aligned} T(n) &= n - 1 + \sum_{i=1}^{n} p_i (T(i-1) + T(n-i)) \\ &\geq n - 1 + n \cdot g(n) - n \cdot g(n) \cdot \sum_{i=1}^{n} p_i (1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2}) \end{aligned}$$

The induction hypothesis follows if $g(n) \leq \left(\frac{n}{n-1} \cdot \sum_{i=1}^{n} p_i (1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2}) \right)^{-1}$ □

The lower bound, Theorem 3, can be given using the entropy function. This shows that up to a logarithmic factor we yield matching upper and lower bounds.

Theorem 4 *For any sequence of probability distributions* (P_n) *it holds that* $T(n) \geq c \cdot g(n) \cdot n$*, for some constant* $c > 0$*, if* g *satisfies the two conditions*

$$g(n) \leq \left(\frac{n}{n-1} \sum_{i=1}^{n} p_{i,n} H\left(\frac{i}{n+1} \right) \right)^{-1} \qquad \textit{and} \qquad \frac{g(i)}{g(n)} \geq \frac{i}{n} \quad \textit{for} \quad 0 \leq i \leq n.$$

Proof. We follow the proof of Theorem 3. For $n \geq 2$

$$\begin{aligned} T(i-1) + T(n-i) &\geq n \cdot g(n) \cdot \left(\frac{(i-1)^2}{n^2} + \frac{(n-i)^2}{n^2} \right) \\ &= n \cdot g(n) \cdot \left(\frac{(i-1)^2}{n^2} + \frac{(n-i)^2}{n^2} + H\left(\frac{i}{n+1} \right) \right) - n \cdot g(n) \cdot H\left(\frac{i}{n+1} \right) \\ &\geq n \cdot g(n) - n \cdot g(n) \cdot H\left(\frac{i}{n+1} \right). \end{aligned}$$

The last inequality follows from Lemma 5 below. Therefore,

$$\begin{aligned} T(n) &= n-1+\sum_{i=1}^{n} p_{i,n}(T(i-1)+T(n-i)) \\ &\geq n-1+n\cdot g(n)-n\cdot g(n)\cdot\sum_{i=1}^{n} p_{i,n} H\left(\frac{i}{n+1}\right). \end{aligned}$$

Then the induction hypothesis follows if $g(n) \leq \left(\frac{n}{n-1}\sum_{i=1}^{n} p_{i,n} H\left(\frac{i}{n+1}\right)\right)^{-1}$. □

Lemma 5 *For integers* $n \geq 5$ *and* i *with* $0 \leq i \leq n$,

$$\frac{(i-1)^2}{n^2}+\frac{(n-i)^2}{n^2}+H\left(\frac{i}{n+1}\right) \geq 1.$$

Proof. We use the known inequalities $-\ln(1-x) \geq x$ resp. $-\log_2(1-x) \geq \frac{x}{\ln 2}$, that hold for $0 \leq x \leq 1$. So we get

$$\begin{aligned} &\frac{(i-1)^2}{n^2}+\frac{(n-i)^2}{n^2}+H\left(\frac{i}{n+1}\right) \\ =\ & \frac{i^2-2i+1+n^2-2in+i^2}{n^2}-\frac{i}{n+1}\log_2\frac{i}{n+1}-\left(1-\frac{i}{n+1}\right)\log_2\left(1-\frac{i}{n+1}\right) \\ =\ & \frac{2i^2-2i+1+n^2-2in}{n^2}-\frac{i}{n+1}\log_2\left(1-\frac{n-i+1}{n+1}\right)-\frac{n-i+1}{n+1}\log_2\left(1-\frac{i}{n+1}\right) \\ \geq\ & \frac{2i^2-2i+1+n^2-2in}{n^2}+\left(\frac{i}{n+1}\cdot\frac{n-i+1}{n+1}+\frac{n-i+1}{n+1}\cdot\frac{i}{n+1}\right)/\ln 2 \\ \geq\ & \frac{2i^2-2i+1+n^2-2in+2in-2i^2+2i}{n^2}=\frac{n^2+1}{n^2}\geq 1 \end{aligned}$$

For the second last inequality, we use that $(n+1)^2\ln 2 \leq n^2$ for $n \geq 5$ and set $n_0 = 5$. □

Remark: Actually, the Lemma holds for $n \geq 1$, not only for $n \geq 5$. The remaining 14 cases $(n,i) = (1,0),(1,1),\ldots,(4,4)$ can be checked by computer.

4 Distributions with bounded entropy

The uniform distribution on $[1,n] = \{1,\ldots,n\}$ has maximal entropy. In this section we consider distributions which have bounded entropy.

Uniform distributions on a subset of $\{1,\ldots,n\}$

First we consider distributions with positive probability on subsets of $[1,n]$. Let $t(n) = o(n)$ be a time constructible monotone (increasing) function. Define a distribution $P = (p_1,\ldots,p_n)$ such that

$$p_i = \begin{cases} 1/t(n), & \text{if rank } a_i \leq t(n)/2 \\ 1/t(n), & \text{if rank } a_i > n-t(n)/2 \\ 0, & \text{otherwise} \end{cases}$$

That is, we choose the pivot element randomly using a uniform distribution among only the worst $t(n)$ array elements.
Now $\sum_{i=1}^{n} p_i H\left(i/(n+1)\right)$ resp. $\sum_{i=1}^{n} p_i \cdot H(i/n)$ are bounded as follows:

$$\begin{aligned}\sum_{i=1}^{n} p_i H\left(\frac{i}{n+1}\right) &\leq \frac{t(n)}{2n}\log\left(n+1\right),\\ \sum_{i=1}^{n} p_i H\left(\frac{i}{n}\right) &\geq \frac{t(n)}{4n}\log\left(\frac{2n}{t(n)}\right)\end{aligned}$$

This gives $T(n) \leq n\log(n)\cdot\frac{4n}{t(n)}$ as an upper bound and $T(n) \geq \frac{2n^2}{t(n)\log n}$ as a lower bound.
Proof. An upper bound $T(n) \leq g(n)\cdot n\cdot\log_2 n$ can be estimated as follows.

$$\begin{aligned}\sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right) &= 2\sum_{i=1}^{t(n)/2}\frac{1}{t(n)}\cdot H\left(\frac{i}{n}\right) \quad = \quad \frac{2}{t(n)}\sum_{i=1}^{t(n)/2} H\left(\frac{i}{n}\right)\\ &= \frac{2}{t(n)}\sum_{i=1}^{t(n)/2} -\left(\frac{i}{n}\log\left(\frac{i}{n}\right)+\frac{n-i}{n}\log\left(\frac{n-i}{n}\right)\right)\\ &\geq \frac{2}{t(n)}\sum_{i=1}^{t(n)/2}\frac{i}{n}\log\left(\frac{n}{i}\right) \quad \geq \quad \frac{2}{n\cdot t(n)}\log\left(\frac{2n}{t(n)}\right)\sum_{i=1}^{t(n)/2} i\\ &\geq \frac{2}{n\cdot t(n)}\log\left(\frac{2n}{t(n)}\right)\frac{(t(n)/2)\cdot(t(n)/2+1)}{2}\\ &\geq \frac{t(n)}{4n}\log\left(\frac{2n}{t(n)}\right).\end{aligned}$$

With

$$g(n) = \frac{4n}{t(n)\log(2n/t(n))}$$

it follows from Theorem 1 that

$$T(n) \leq \frac{4n^2}{t(n)}\cdot\frac{\log_2 n}{\log_2(2n/t(n))}$$

In the same way the lower bound can be calculated:

$$\begin{aligned}
\sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n+1}\right) &= 2\sum_{i=1}^{t(n)/2} \frac{1}{t(n)} \cdot H\left(\frac{i}{n+1}\right) \quad = \quad \frac{2}{t(n)} \sum_{i=1}^{t(n)/2} H\left(\frac{i}{n+1}\right) \\
&= \frac{2}{t(n)} \sum_{i=1}^{t(n)/2} -\left(\frac{i}{n+1}\log\left(\frac{i}{n+1}\right) + \frac{n-i+1}{n+1}\log\left(\frac{n-i+1}{n+1}\right)\right) \\
&\leq \frac{2}{t(n)} \sum_{i=1}^{t(n)/2} 2 \cdot \frac{i}{n+1}\log\left(\frac{n+1}{i}\right) \\
&= \frac{4}{(n+1)t(n)} \sum_{i=1}^{t(n)/2} i\log\left(\frac{n+1}{i}\right) \\
&= \frac{4}{(n+1)t(n)} \left(\sum_{i=1}^{t(n)/2} i\log(n+1) - \sum_{i=1}^{t(n)/2} i\log i\right) \\
&\leq \frac{4}{(n+1)t(n)} \left(\sum_{i=1}^{t(n)/2} i\log(n+1) - \sum_{i=1}^{t(n)/2} i(\log(t(n)/2)-1)\right) \\
&\leq \frac{t(n)+1}{2(n+1)}\left(\log(n+1) - \log t(n) + 2\right) \\
&\leq \frac{t(n)+2}{2(n+1)}\log\left(\frac{4(n+1)}{t(n)}\right)
\end{aligned}$$

where we use that $\sum_{i=1}^{t(n)/2} i\log i \leq \sum_{i=1}^{t(n)/2} i(\log(t(n)/2)-1)$ (see Appendix, Lemma 10). With the function

$$g(n) = \frac{2(n+1)}{(t(n)+1)\log\left(\frac{4(n+1)}{t(n)}\right)} \quad ,$$

we receive a lower bound of

$$T(n) \geq \frac{2n(n+1)}{(t(n)+1)\log\left(\frac{4(n+1)}{t(n)}\right)} = \Omega\left(\frac{n^2}{t(n)\log\left(\frac{4n}{t(n)}\right)}\right) \quad .$$

□

Min-Entropy

A distribution $(p_1, \ldots, p_n)$ has *min-entropy* k (cf. [Lu]) if $p_i \leq 2^{-k}$ for all i. Let $P = (p_1, \ldots, p_n)$ be a distribution with min-entropy k. Then we get

$T(n) \leq \frac{4n^2}{2^k}$ as an upper bound and $T(n) \geq \frac{2n^2}{2^k \log n}$ as a lower bound.

Proof.

$$\begin{aligned}
\sum_{i=1}^{n} p_i \cdot H(i/n) &\geq 2\sum_{i=1}^{2^k/2} \frac{1}{2^k} \cdot H(i/n) \\
&\geq \ldots \text{(same as above, with } t(n) = 2^k\text{)} \\
&\geq \frac{2^k}{4n}\log\left(\frac{2n}{2^k}\right),
\end{aligned}$$

and

$$\begin{aligned}\sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n+1}\right) &\leq 2\sum_{i=1}^{2^k/2} \frac{1}{2^k} \cdot H\left(\frac{i}{n+1}\right)\\ &\leq \frac{2^k+1}{2(n+1)} \log\left(\frac{2(n+1)}{2^k}\right)\end{aligned}$$

and thus

$$T(n) \leq \frac{4n^2}{2^k} \cdot \frac{\log_2 n}{\log_2(2n/2^k)}$$

and

$$T(n) \geq \frac{2n(n+1)}{(2^k+1)\log\left(\frac{2(n+1)}{2^k}\right)}$$

□

So, for min-entropy 0 (this includes the deterministic case) we get

$$T(n) \leq \frac{4n^2}{1} \cdot \frac{\log_2 n}{\log_2(2n)} = 4n^2 \frac{\log_2 n}{\log_2 n + 1} \leq 4n^2$$

and

$$T(n) \geq \frac{n(n+1)}{\log\left(2(n+1)\right)} \geq \frac{n^2}{\log\left(n+1\right)+1}$$

and for min-entropy $\log_2 n$ (all pivot elements are equally distributed), we have

$$T(n) \leq \frac{4n^2}{n} \cdot \frac{\log_2 n}{\log_2 2} = 4n \log_2 n \ \ .$$

Bounds for geometric distributions

We consider the case that pivot elements are selected using a geometric distribution. The probability of picking an element with rank i as pivot is given by $p_i = q^{i-1}(1-q)$. More generally, we allow the geometric distribution to depend on the size n of the array, i.e., we define (P_i) using $q := 1 - \frac{1}{f(i)}$ for some (time constructible monotone) function f.

To estimate a lower bound on the number of comparisons, we use Theorem 3 and estimate $\sum_{i=1}^{n} p_i \left(1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2}\right) \leq \frac{cf(n)}{n}$, for a constant c.

Proof. Using the fact that

$$q^i = \left(1 - \frac{1}{f(n)}\right)^i = \left(1 - \frac{1}{f(n)}\right)^{f(n) \cdot \frac{i}{f(n)}} \leq e^{-\frac{i}{f(n)}} \ \ ,$$

it follows that

$$\begin{aligned}
&\sum_{i=1}^{n} p_i \left(1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2}\right) \\
&\quad = \frac{1}{q}\sum_{i=1}^{n} q^i(1-q)\left(1 - \frac{(i-1)^2}{n^2} - \frac{(n-i)^2}{n^2}\right) \\
&\quad = \frac{1}{qn^2}\sum_{i=1}^{n} q^i(1-q)\left(2ni + 2i - 2i^2 - 1\right) \\
&\quad \leq \frac{1}{qn^2}\sum_{i=1}^{n} q^i(1-q)\left(2ni + 2i\right) \\
&\quad = \frac{1}{\left(1 - \frac{1}{f(n)}\right)n^2}\sum_{i=1}^{n}\left(1 - \frac{1}{f(n)}\right)^i \frac{1}{f(n)}\left(2ni + 2i\right) \\
&\quad = \frac{(2n+2)f(n)}{n^2(f(n)-1)}\sum_{i=1}^{n}\left(1 - \frac{1}{f(n)}\right)^i \frac{i}{f(n)}
\end{aligned}$$

We split the sum and see that for $k = 0, 1, 2, \ldots$

$$\begin{aligned}
&\sum_{i=kf(n)+1}^{(k+1)f(n)} \left(1 - \frac{1}{f(n)}\right)^i \frac{i}{f(n)} \\
&\quad \leq \sum_{i=kf(n)+1}^{(k+1)f(n)} e^{-\frac{i}{f(n)}+\ln\frac{i}{f(n)}} = \sum_{j=1}^{f(n)} e^{-\frac{kf(n)+j}{f(n)}+\ln\frac{kf(n)+j}{f(n)}} \\
&\quad \leq \sum_{j=1}^{f(n)} e^{-k-\frac{j}{f(n)}+\ln(k+1)} = e^{-k+\ln(k+1)}\sum_{j=1}^{f(n)} e^{-\frac{j}{f(n)}} \\
&\quad \leq e^{-k+\ln(k+1)} \cdot f(n).
\end{aligned}$$

Then we get

$$\begin{aligned}
&\frac{(2n+2)f(n)}{n^2(f(n)-1)}\sum_{i=1}^{n}\left(1 - \frac{1}{f(n)}\right)^i \frac{i}{f(n)} \\
&\quad = \frac{(2n+2)f(n)}{n^2(f(n)-1)}\sum_{k=0}^{\lceil n/f(n)\rceil}\sum_{i=kf(n)+1}^{(k+1)f(n)}\left(1 + \frac{1}{f(n)}\right)^i \cdot \frac{i}{f(n)} \\
&\quad \leq \frac{(2n+2)f(n)}{n^2(f(n)-1)}\sum_{k=0}^{\lceil n/f(n)\rceil} e^{-k+\ln(k+1)} \cdot f(n) \\
&\quad \leq \frac{(2n+2)f(n)^2}{n^2(f(n)-1)}\sum_{k=0}^{\infty}\frac{k+1}{e^k} \\
&\quad \approx \frac{cf(n)}{n} \text{ for a constant } c.
\end{aligned}$$

Using Theorem 3, we get a lower bound of $c'n^2/f(n)$ for the running time of QuickSort. □

To get an upper bound for geometric distributions we estimate

$$\sum_{i=1}^{n} p_i H\left(\frac{i}{n}\right) \geq \frac{\log n \cdot (f(n) - n \cdot e^{-n/f(n)})}{n}$$

which gives $T(n) \leq \frac{n^2}{f(n)}$ as upper bound, if $f(n) = o(n)$.
Proof.

$$\begin{aligned}
\sum_{i=1}^{n} p_i H\left(\frac{i}{n}\right) &= \frac{1-q}{q} \sum_{i=1}^{n} q^i H\left(\frac{i}{n}\right) \\
&= \frac{1-q}{q} \sum_{i=1}^{n} q^i \left(\frac{i}{n} \log \frac{n}{i} + \frac{n-i}{n} \log \frac{n}{n-i}\right) \\
&\geq \frac{1-q}{q} \sum_{i=1}^{n} q^i \left(\frac{i}{n} \log \frac{n}{i}\right) \\
&\geq \frac{1-q}{qn} \log n \sum_{i=1}^{n-1} q^i \cdot i \\
&= \frac{1-q}{qn} \log n \left(\frac{q^n(nq-n-q)}{(1-q)^2} + \frac{q}{(1-q)^2}\right) \\
&= \frac{\log n}{n} \left(\frac{q^{n-1}(nq-n-q)}{1-q} + \frac{1}{1-q}\right)
\end{aligned}$$

We again set $q := 1 - \frac{1}{f(n)}$ to obtain

$$\begin{aligned}
\sum_{i=1}^{n} p_i H\left(\frac{i}{n}\right) &= \frac{\log n}{n} \left(\frac{\left(1-\frac{1}{f(n)}\right)^{n-1}\left(n\left(1-\frac{1}{f(n)}\right) - n - 1 + \frac{1}{f(n)}\right)}{\frac{1}{f(n)}} + \frac{1}{\frac{1}{f(n)}}\right) \\
&= \frac{\log n f(n)}{n} \left((n-1)\left(1-\frac{1}{f(n)}\right)^{n} - n\left(1-\frac{1}{f(n)}\right)^{n-1} + 1\right) \\
&= \frac{\log n f(n)}{n} \left(\left(1-\frac{1}{f(n)}\right)^{n-1}\left((n-1)\left(1-\frac{1}{f(n)}\right) - n\right) + 1\right) \\
&= \frac{\log n f(n)}{n} \left(\left(1-\frac{1}{f(n)}\right)^{n-1}\left(-1-\frac{n-1}{f(n)}\right) + 1\right) \\
&= \frac{\log n f(n)}{n} \left(1 - \left(1-\frac{1}{f(n)}\right)^{n-1}\left(1+\frac{n-1}{f(n)}\right)\right) \\
&\geq \frac{\log n f(n)}{n} \left(1 - e^{-\frac{n-1}{f(n)}} \cdot \frac{2n}{f(n)}\right) \\
&\geq \frac{c \log n f(n)}{n} \quad \text{for some constant } c > 0 \text{ if } f(n) = o(n)
\end{aligned}$$

So we have an upper bound for the worst-case running time of $T(n) \leq \frac{cn^2}{f(n)}$ for some constant $c > 0$.

5 The δ-random source

A general model of a random bit generator is the δ-random-source. Since the bias of each bit is a function of the previous output, it can be applied as an adversary argument and is particularly suited for worst-case analysis. See also [Pa, SV, AR].

Definition 6 (See [AR]) *A δ-random-source is a random bit generator. Its bias may depend on the bits it has previously output, but the probability to output "1" must be in the range $[\delta, 1-\delta]$. Therefore, it has an internal state $\omega \in \{0,1\}^*$, denoting its previously output bits.*

To obtain a random number X in the range $1, \ldots, n$ from the δ-random-source, we output $\lceil \log n \rceil$ bits and interpret them as a number Y. Then, we set $X := (Y \bmod n) + 1$.

Lemma 7 (See [As]) *For each p with $0 < p < \frac{1}{2}$, there exists a constant c, such that for all $n \in \mathbb{N}$:*

$$c(p) \cdot \frac{2^{H(p) \cdot n}}{\sqrt{n}} \leq \sum_{j=0}^{\lfloor np \rfloor} \binom{n}{j} \leq 2^{H(p) \cdot n} \quad .$$

Lemma 8 (deMoivre-Laplace Limit Theorem) *For each p, $0 < p < 1$,*

$$\lim_{n \to \infty} \sum_{k=0}^{\lfloor pn \rfloor} \binom{n}{k} p^k (1-p)^{n-k} = \frac{1}{2}$$

Proof. Let $S_{n,p}$ be a binomially distributed random variable with parameters n and p. The normalized binomial distribution can be approximated by the normal distribution Φ, so

$$\begin{aligned}
\lim_{n \to \infty} \sum_{k=0}^{\lfloor pn \rfloor} \binom{n}{k} p^k (1-p)^{n-k} &= \lim_{n \to \infty} Pr[S_{n,p} \leq pn] \\
&= \lim_{n \to \infty} Pr\left[\frac{S_{n,p} - np}{\sqrt{np(1-p)}} \leq 0\right] \\
&= \Phi(0) \\
&= \frac{1}{2}
\end{aligned}$$

Theorem 9 *For each δ-random-source, $0 < \delta < \frac{1}{2}$, there exists $n_0 \in \mathbb{N}$, such that for each $n > n_0$, and each permutation π, Theorem 1 can be applied with*

$$g(n) = c(\delta) \cdot \frac{1}{\sqrt{\log n}} \cdot n^{1-H(\delta)},$$

where the random bits are produced by a δ-random-source (modulo n) and $c(\delta)$ is a constant that depends on δ.

Proof. From the symmetry and monotony of the entropy function it follows that for each s

$$\sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right) \geq \left(1 - \sup_{\pi, \tilde{\omega}} \sum_{j=1}^{s-1} p_j\right) \cdot H\left(\frac{s}{2n}\right), \tag{1}$$

where p_j depends on π and on the internal state $\tilde{\omega}$ of the random source.

Now we examine the two factors on the righthand side of (1) separately. We set

$$k := \lceil \log n \rceil$$

and

$$s := \frac{1}{2} \sum_{j=0}^{\lfloor \delta k \rfloor} \binom{k}{j}.$$

Since

$$p_j = \begin{cases} Pr[Y = \pi(j)], & n + \pi(j) \geq 2^k \\ Pr[Y = \pi(j)] + Pr[Y = \pi(j) + n] & \text{otherwise} \end{cases},$$

we get for the first factor of (1)

$$\begin{aligned} \sup_{\pi,\tilde{\omega}} \sum_{j=1}^{s-1} p_j &\leq \sup_{\tilde{\omega}} \max_{M \subseteq \{0,1\}^k, |M| = 2s} Pr[Y \in M] \\ &\leq \sum_{j=0}^{\lfloor \delta k \rfloor} \binom{k}{j} \delta^j (1-\delta)^{k-j}. \end{aligned}$$

Here we use the result from [AR], that the maximum probability of hitting a set of a certain size can be achieved by an "extreme" δ-random-source that always outputs "0" with probability δ.

Since by Lemma 8

$$\lim_{k \to \infty} \sum_{j=0}^{\lfloor \delta k \rfloor} \binom{k}{j} \delta^j (1-\delta)^{k-j} = \frac{1}{2},$$

there exists some constant $c'(\delta)$, so that

$$\sup_{\pi,\tilde{\omega}} \sum_{j=1}^{s-1} p_j \leq c'(\delta).$$

Now we consider the right factor of the equation above. We use the monotony of $H(x)$ on the intervall $[0, \frac{1}{2}]$ and Lemma 7:

$$H\left(\frac{s}{2n}\right) \geq H\left(\frac{s}{2^{k+1}}\right) \geq H\left(c_1(\delta) \cdot \frac{2^{(H(\delta)-1)k}}{4\sqrt{k}}\right)$$

We consider $\delta < \frac{1}{2}$ (so that $H(\delta) < 1$) and use that $H(x) \geq -x \log x$. So we get

$$H\left(\frac{s}{2n}\right) \geq c_1(\delta) \cdot \frac{2^{(H(\delta)-1)k}}{4\sqrt{k}} \cdot \left[(1 - H(\delta))k - \log \frac{c_1(\delta)}{4\sqrt{k}}\right].$$

For k big enough ($k > k_0$ corresponds to $n > n_0$), there is a constant $c''(\delta)$ so that

$$H\left(\frac{s}{2n}\right) \geq c''(\delta) \cdot \sqrt{k} \cdot 2^{(H(\delta)-1)k}.$$

Combining the results, there is a $n_0 \in \mathbb{N}$ and a $c^*(\delta)$, such that for all $n \geq n_0$, and all permutations π on $\{0, \ldots, n-i\}$ and all states $\tilde{\omega} \in \{0,1\}^*$ of the generator the following holds:

$$\begin{aligned} \sum_{i=1}^{n} p_i \cdot H\left(\frac{i}{n}\right) &\geq c^*(\delta) \cdot \sqrt{\lceil \log n \rceil} \cdot 2^{(H(\delta)-1)\lceil \log n \rceil} \\ &\geq \frac{1}{c(\delta)} \cdot \sqrt{\log n} \cdot n^{H(\delta)-1}, \end{aligned}$$

which leads to the expected running time of $T(n) \leq c(\delta) \cdot n^{2-H(\delta)} \cdot \sqrt{\log n}$. □

References

[AR] Noga Alon, Michael O. Rabin: Biased coins and randomized algorithms. In: F.P. Preparata, S. Micali (eds): *Advances in Computing Research 5*. JAI Press, 1989, pages 499–507.

[As] R.B. Ash: *Information Theory*. Dover 1965.

[Ho] C.A.R. Hoare: Quicksort. *Computer Journal*, 5(1): 10–15, 1962.

[Kn] Donald Knuth: *The Art of Computer Programming*. Vol 3: Sorting and Searching. Addison-Wesley, 1973.

[KR] H.J. Karloff, P. Raghavan: Randomized algorithms and pseudorandom numbers. *Journal of the Association for Computing Machinery* 40 (1993) 454–476.

[Li] Beatrice List: *Probabilistische Algorithmen und schlechte Zufallszahlen*. Doctoral Dissertation, Universit¨at Ulm, 1999.

[Lu] Michael Luby: *Pseudorandomness and Cryptographic Applications*. Princeton University Press, 1996.

[Pa] C. H. Papadimitriou: *Computational Complexity*. Addison-Wesley, 1994 (pages 259ff)

[SF] Robert Sedgewick, Philippe Flajolet: *Analysis of Algorithms*. Addison-Wesley, 1996.

[SV] M. Santha, U. V. Vazirani: *Generating quasi-random sequences from slightly random sources*. Proceedings of the 25th IEEE

[To] Martin Tompa: *Probabilistic Algorithms and Pseudorandom Generators*. Lecture Notes, 1991.

Appendix

Lemma 10 *For* $0 \leq i \leq n$

$$\sum_{i=1}^{n} i \log_2 i \geq \sum_{i=1}^{n} i(\log_2 n - 1) \ .$$

Proof. Let $S(n) := \sum_{i=1}^{n} i \log_1 i$. We prove the lemma by induction.

$$\begin{aligned}
S(n) &= \sum_{i=1}^{n/2} i \log_2 i + \sum_{i=n/2+1}^{n} i(\log_2 n - 1 + 1 + \log_2 (i/n)) \\
&\geq \sum_{i=1}^{n/2} i(\log_2(n/2) - 1) + \sum_{i=n/2+1}^{n} i(\log_2 n - 1) + \sum_{i=n/2+1}^{n} i(1 + \log_2 (i/n)) \\
&= \sum_{i=1}^{n} i(\log_2 n - 1) - \sum_{i=1}^{n/2} i + \sum_{i=n/2+1}^{n} i(1 + \log_2 (i/n)) \\
&\geq \sum_{i=1}^{n} i(\log_2 n - 1) - \sum_{i=1}^{n/2} i + \sum_{i=n/2+1}^{n} i(2i/n - 1) \\
&= \sum_{i=1}^{n} i(\log_2 n - 1) - \sum_{i=1}^{n} i + 2 \sum_{i=n/2+1}^{n} i^2/n \\
&= \sum_{i=1}^{n} i(\log_2 n - 1) + \frac{1}{n} \sum_{i=1}^{n/2} \big(2(n/2 + i)^2 - ni - n(n/2 + i)\big) \\
&= \sum_{i=1}^{n} i(\log_2 n - 1) + \frac{1}{n} \sum_{i=1}^{n/2} 2i^2 \\
&\geq \sum_{i=1}^{n} i(\log_2 n - 1)
\end{aligned}$$

□

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'