THE NUMBER OF BIT COMPARISONS USED BY QUICKSORT: AN AVERAGE-CASE ANALYSIS

JAMES ALLEN FILL AND SVANTE JANSON

Abstract. The analyses of many algorithms and data structures (such as digital search trees) for searching and sorting are based on the representation of the keys involved as bit strings and so count the number of bit comparisons. On the other hand, the standard analyses of many other algorithms (such as `Quicksort`) are performed in terms of the number of key comparisons. We introduce the prospect of a fair comparison between algorithms of the two types by providing an average-case analysis of the number of bit comparisons required by `Quicksort`. Counting bit comparisons rather than key comparisons introduces an extra logarithmic factor to the asymptotic average total. We also provide a new algorithm, "`BitsQuick`", that reduces this factor to constant order by eliminating needless bit comparisons.

1. Introduction and summary

Algorithms for sorting and searching (together with their accompanying analyses) generally fall into one of two categories: either the algorithm is regarded as comparing items pairwise irrespective of their internal structure (and so the analysis focuses on the number of comparisons), or else it is recognized that the items (typically numbers) are represented as bit strings and that the algorithm operates on the individual bits. Typical examples of the two types are `Quicksort` and digital search trees, respectively; see [15].

In this paper—a substantial expansion of the extended abstract [7]—we take a first step towards bridging the gap between the two points of view, in order to facilitate run-time comparisons across the gap, by answering the following question posed many years ago by Bob Sedgewick [personal communication]: What is the bit complexity of `Quicksort`? (For a discussion of related work that has transpired in the time between [7] and this paper, see Remark 1.6 at the end of this section.)

More precisely, we consider `Quicksort` (see Section 2 for a review) applied to n distinct keys (numbers) from the interval $(0, 1)$. Many authors (Knuth [15], Régnier [18], Rösler [20], Knessl and Szpankowski [14], Fill and Janson [5] [6], Neininger and Rüschendorf [17], and others) have studied K_n, the (random) number of key comparisons performed by the algorithm. This is a natural measure of the cost (run-time) of the algorithm, if each comparison has the same cost. On the other hand, if comparisons are done by scanning the bit representations of the numbers, comparing their bits one by one, then the cost of comparing two keys is determined by the number of bits compared until a difference is found. We call this

Date: February 10, 2012; revised May 27, 2012.

Research of the first author supported by NSF grants DMS-0104167 and DMS-0406104 and by the Acheson J. Duncan Fund for the Advancement of Research in Statistics.

number the number of *bit comparisons* for the key comparison, and let B_n denote the total number of bit comparisons when n keys are sorted by `Quicksort`.

We assume that the keys $X_1, \dots, X_n$ to be sorted are independent random variables with a common continuous distribution F over $(0,1)$. It is well known that the distribution of the number K_n of key comparisons does not depend on F. This invariance clearly fails to extend to the number B_n of bit comparisons, and so we need to specify F.

For simplicity, we study mainly the case that F is the uniform distribution, and, throughout, the reader should assume this as the default. But we also give a result valid for a general absolutely continuous distribution F over $(0,1)$ (subject to a mild integrability condition on the density).

In this paper we focus on the mean of B_n. One of our main results is the following Theorem 1.1, the concise version of which is the asymptotic equivalence

$$\mathbf{E}\, B_n \sim n(\ln n)(\lg n) \text{ as } n \to \infty.$$

Throughout, we use ln (respectively, lg) to denote natural (resp., binary) logarithm, and use log when the base doesn't matter (for example, in remainder estimates). The symbol $\doteq$ is used to denote approximate equality, and $\gamma \doteq 0.57722$ is Euler's constant.

Theorem 1.1. *If the keys $X_1, \dots, X_n$ are independent and uniformly distributed on $(0,1)$, then the number B_n of bit comparisons required to sort these keys using* `Quicksort` *has expectation given by the following exact and asymptotic expressions:*

$$\mathbf{E}\, B_n = 2\sum_{k=2}^{n} (-1)^k \binom{n}{k} \frac{1}{(k-1)k[1-2^{-(k-1)}]} \tag{1.1}$$

$$= n(\ln n)(\lg n) - c_1 n \ln n + c_2 n + \pi_n n + O(\log n), \tag{1.2}$$

where, with $\beta := 2\pi/\ln 2$,

$$c_1 := \frac{1}{\ln 2}(4 - 2\gamma - \ln 2) \doteq 3.105,$$

$$c_2 := \frac{1}{\ln 2}\left[\frac{1}{6}(6-\ln 2)^2 - (4-\ln 2)\gamma + \frac{\pi^2}{6} + \gamma^2\right] \doteq 6.872,$$

and

$$\pi_n := \sum_{k \in \mathbb{Z}:\, k \neq 0} \frac{i}{\pi k(-1-i\beta k)} \Gamma(-1-i\beta k) n^{i\beta k} \tag{1.3}$$

is periodic in $\lg n$ with period 1 and amplitude smaller than 5×10^{-9}.

Small periodic fluctuations as in Theorem 1.1 come as a surprise to newcomers to the analysis of algorithms but in fact are quite common in the analysis of digital structures and algorithms; see, for example, Chapter 6 in [16].

For our further results, it is technically convenient to assume that the number of keys is no longer fixed at n, but rather Poisson distributed with mean λ and independent of the values of the keys. (In this paper, we shall not deal with the "de-Poissonization" that would be needed to transfer results back to the fixed-n

model.) In obvious notation, the Poissonized version of (1.1)–(1.2) is

$$\mathbf{E}\,B(\lambda) = 2\sum_{k=2}^{\infty}(-1)^k \frac{\lambda^k}{k!} \times \frac{1}{(k-1)k[1-2^{-(k-1)}]} \tag{1.4}$$

$$= \lambda(\ln\lambda)(\lg\lambda) - c_1\lambda\ln\lambda + c_2\lambda + \pi_\lambda\lambda + O(\log\lambda) \quad \text{as } \lambda \to \infty, \tag{1.5}$$

with π_λ as in (1.3). The exact formula follows immediately from (1.1), and the asymptotic formula is established in Section 5 as Proposition 5.4. We will also see (Proposition 5.6) that $\mathbf{Var}\,B(\lambda) = O(\lambda^2)$, so $B(\lambda)$ is concentrated about its mean. Since the number $K(\lambda)$ of key comparisons is likewise concentrated about its mean $\mathbf{E}\,K(\lambda) \sim 2\lambda\ln\lambda$ for large λ (see Lemmas 5.1 and 5.3), it follows that

$$\frac{2}{\lg\lambda} \times \frac{B(\lambda)}{K(\lambda)} \to 1 \quad \text{in probability as } \lambda \to \infty. \tag{1.6}$$

In other words, about $\frac{1}{2}\lg\lambda$ bits are compared per key comparison.

Remark 1.2. Further terms can be obtained in (1.2) and (1.5) by the methods used in the proofs below. In particular, the $O(\log\lambda)$ in (1.5) can be refined to

$$-2\log\lambda - c_4 + O(\lambda^{-M})$$

for any fixed M, with

$$c_4 := 4\ln 2 + 2 + 2\gamma \doteq 5.927.$$

For non-uniform distribution F, we have the same leading term for the asymptotic expansion of $\mathbf{E}\,B(\lambda)$, but the second-order term is larger. (Throughout, $\ln_+$ denotes the positive part of the natural logarithm function. We denote the uniform distribution by unif.)

Theorem 1.3. *Let $X_1, X_2, \ldots$ be independent with a common distribution F over $(0,1)$ having density f, and let N be independent and Poisson with mean λ. If $\int_0^1 f(\ln_+ f)^4 < \infty$, then the expected number of bit comparisons, call it $\mu_f(\lambda)$, required to sort the keys $X_1, \ldots, X_N$ using* `Quicksort` *satisfies*

$$\mu_f(\lambda) = \mu_{\text{unif}}(\lambda) + 2H(f)\lambda\ln\lambda + o(\lambda\log\lambda)$$

as $\lambda \to \infty$, where $H(f) := \int_0^1 f\lg f \geq 0$ is the entropy (in bits) of the density f.

In applications, it may be unrealistic to assume that a specific density f is known. Nevertheless, even in such cases, Theorem 1.3 may be useful since it provides a measure of the robustness of the asymptotic estimate in Theorem 1.1.

Bob Sedgewick (among others who heard us speak on the material of this paper) suggested that the number of bit comparisons for `Quicksort` might be reduced substantially by not comparing bits that have to be equal according to the results of earlier steps in the algorithm. In the final section (Theorem 7.1), we note that this is indeed the case: For a fixed number n of keys, the average number of bit comparisons in the improved algorithm (which we dub "`BitsQuick`") is asymptotically equivalent to $2(1 + \frac{3}{2\ln 2})n\ln n$, only a constant ($\doteq 3.2$) times the average number of key comparisons [see (2.2)]. A related algorithm is the digital version of Quicksort by Roura [21]; it too requires $\Theta(n\log n)$ bit comparisons (we do not know the exact constant factor).

We may compare our results to those obtained for radix-based methods, for example radix exchange sorting, see [15, Section 5.2.2]. This method works by bit inspections, that is, by comparisons to constant bits, rather than by pairwise comparisons. In the case of n uniformly distributed keys, radix exchange sorting uses asymptotically $n \lg n$ bit inspections. Since radix exchange sorting is designed so that the number of bit inspections is minimal, it is not surprising that our results show that `Quicksort` uses more bit comparisons. More precisely, Theorem 1.1 shows that `Quicksort` uses about $\ln n$ times as many bit comparisons as radix exchange sorting. For `BitsQuick`, this is reduced to a small constant factor. This gives us a measure of the cost in bit comparisons of using these algorithms; `Quicksort` is often used because of other advantages, and our results open the possibility of seeing when they outweigh the increase in bit comparisons.

In Section 2 we review `Quicksort` itself and basic facts about the number K_n of key comparisons. In Section 3 we derive the exact formula (1.1) for $\mathbf{E}\, B_n$, and in Section 4 we derive the asymptotic expansion (1.2) from an alternative exact formula that is somewhat less elementary than (1.1) but much more transparent for asymptotics. In the transitional Section 5 we establish certain basic facts about the moments of $K(\lambda)$ and $B(\lambda)$ in the Poisson case with uniformly distributed keys, and in Section 6 we use martingale arguments to establish Theorem 1.3 for the expected number of bit comparisons for Poisson(λ) draws from a general density f. Finally, in Section 7 we study the improved `BitsQuick` algorithm discussed in the preceding paragraph.

Remark 1.4. The results can be generalized to bases other than 2. For example, base 256 would give corresponding results on the "byte complexity".

Remark 1.5. Cutting off and sorting small subfiles differently would affect the results in Theorems 1.1 and 1.3 by $O(n \log n)$ and $O(\lambda \log \lambda)$ only. In particular, the leading terms would remain the same.

Remark 1.6. In comparison with the extended abstract [7], new in this expanded treatment are Remark 5.2, Propositions 5.4 and 5.7, and Lemma 6.2, together with complete proofs of Theorem 1.3, Lemmas 5.1 and 5.3, and Remark 6.3. Section 7 has been substantially revised.

In the time between [7] and the present paper, the following developments have occurred:

- Fill and Nakama [8] followed the same sort of approach as in this paper to obtain certain exact and asymptotic expressions for the number of bit comparisons required by `Quickselect`, a close cousin of `Quicksort`.
- Vallée et al. [22] used analytic-combinatorial methods to extend the results of [7] and [8] by deriving asymptotic expressions for the expected number of symbol comparisons for both `Quicksort` and `Quickselect`. In their work, as in the present paper, the keys are assumed to be independent and identically distributed, but the authors allow for quite general probabilistic models (also known as "sources") for how each key is generated as a symbol string.
- Fill and Nakama [9] obtained, for quite general sources, a limiting distribution for the (suitably scale-normalized) number of symbol comparisons required by `Quickselect`.
- Fill [4] obtained, for quite general sources, a limiting distribution for the (suitably center-and-scale-normalized) number of symbol comparisons required by `Quicksort`.

We were motivated to expand [7] to the present full-length paper in large part because this paper's Lemmas 5.1 and 5.3, and an extension of (the proof of) Proposition 5.7, play key roles in [4].

2. Review: number of key comparisons used by Quicksort

In this section we briefly review certain basic known results concerning the number K_n of key comparisons required by `Quicksort` for a fixed number n of keys uniformly distributed on $(0, 1)$. (See, for example, [6] and the references therein for further details.)

`Quicksort`, invented by Hoare [13], is the standard sorting procedure in `Unix` systems, and has been cited [3] as one of the ten algorithms "with the greatest influence on the development and practice of science and engineering in the 20th century." The `Quicksort` algorithm for sorting an array of n distinct keys is very simple to describe. If $n = 0$ or $n = 1$, there is nothing to do. If $n \geq 2$, pick a key uniformly at random from the given array and call it the "pivot". Compare the other keys to the pivot to partition the remaining keys into two subarrays. Then recursively invoke `Quicksort` on each of the two subarrays.

With $K_0 := 0$ as initial condition, K_n satisfies the distributional recurrence relation

$$K_n \overset{\mathcal{L}}{=} K_{U_n-1} + K^*_{n-U_n} + n - 1, \qquad n \geq 1,$$

where $\overset{\mathcal{L}}{=}$ denotes equality in law (i.e., in distribution), and where, on the right, U_n is distributed uniformly over the set $\{1, \ldots, n\}$, $K^*_j \overset{\mathcal{L}}{=} K_j$ for every j, and

$$U_n; \; K_0, \ldots, K_{n-1}; \; K^*_0, \ldots, K^*_{n-1}$$

are all independent.

Passing to expectations we obtain the "divide-and-conquer" recurrence relation

$$\mathbf{E}\, K_n = \frac{2}{n} \sum_{j=0}^{n-1} \mathbf{E}\, K_j + n - 1,$$

which is easily solved to give

$$\begin{aligned} \mathbf{E}\, K_n &= 2(n+1)H_n - 4n && (2.1)\\ &= 2n \ln n - (4 - 2\gamma)n + 2 \ln n + (2\gamma + 1) + O(1/n). && (2.2) \end{aligned}$$

It is also routine to use a recurrence to compute explicitly the exact variance of K_n. In particular, the asymptotics are

$$\mathbf{Var}\, K_n = \sigma^2 n^2 - 2n \ln n + O(n)$$

where $\sigma^2 := 7 - \frac{2}{3}\pi^2 \doteq 0.4203$. Higher moments can be handled similarly. Further, the normalized sequence

$$\widehat{K}_n := (K_n - \mathbf{E}\, K_n)/n, \qquad n \geq 1,$$

converges in distribution, with convergence of moments of each order, to $\widehat{K}$, where the law of $\widehat{K}$ is characterized as the unique distribution over the real line with vanishing mean that satisfies a certain distributional identity; and the moment generating functions of $\widehat{K}_n$ converge pointwise to that of $\widehat{K}$.

3. Exact mean number of bit comparisons

In this section we establish the exact formula (1.1), repeated here for convenience as (3.1), for the expected number of bit comparisons required by `Quicksort` for a fixed number n of keys uniformly distributed on $(0,1)$:

$$\mathbf{E}\,B_n = 2\sum_{k=2}^{n}(-1)^k\binom{n}{k}\frac{1}{(k-1)k[1-2^{-(k-1)}]}. \tag{3.1}$$

Let $X_1,\dots,X_n$ denote the keys, and $X_{(1)}<\dots<X_{(n)}$ their order statistics. Consider ranks $1\le i<j\le n$. Formula (3.1) follows readily from the following three facts, all either obvious or very well known:

- The event $C_{ij} := \{\text{keys } X_{(i)} \text{ and } X_{(j)} \text{ are compared}\}$ and the random vector $(X_{(i)},X_{(j)})$ are independent.
- $\mathbf{P}(C_{ij}) = 2/(j-i+1)$. [Indeed, C_{ij} equals the event that the first pivot chosen from among $X_{(i)},\dots,X_{(j)}$ is either $X_{(i)}$ or $X_{(j)}$.]
- The joint density $g_{n,i,j}$ of $(X_{(i)},X_{(j)})$ is given by

$$g_{n,i,j}(x,y) = \binom{n}{i-1,1,j-i-1,1,n-j}x^{i-1}(y-x)^{j-i-1}(1-y)^{n-j}. \tag{3.2}$$

Let $b(x,y)$ denote the index of the first bit at which the numbers $x,y\in(0,1)$ differ. (For definiteness we take in this paper the terminating expansion with infinitely many zeros for dyadic rationals in $[0,1)$, but $1 = .111\dots$.) Then

$$\begin{aligned}\mathbf{E}\,B_n &= \sum_{1\le i<j\le n}\mathbf{P}(C_{ij})\int_0^1\int_x^1 b(x,y)\,g_{n,i,j}(x,y)\,dy\,dx \\ &= \int_0^1\int_x^1 b(x,y)\,p_n(x,y)\,dy\,dx,\end{aligned} \tag{3.3}$$

where $p_n(x,y)$ has the definition and interpretation

$$\begin{aligned}p_n(x,y) &:= \sum_{1\le i<j\le n}\mathbf{P}(C_{ij})g_{n,i,j}(x,y)\\ &= \frac{\mathbf{P}(\text{keys in }(x,x+dx)\text{ and }(y,y+dy)\text{ are compared})}{dx\,dy}.\end{aligned}$$

By a routine calculation,

$$\begin{aligned}p_n(x,y) &= \frac{2}{(y-x)^2}\left[(1-(y-x))^n - 1 + n(y-x)\right]\\ &= 2\sum_{k=2}^{n}(-1)^k\binom{n}{k}(y-x)^{k-2},\end{aligned} \tag{3.4}$$

which depends on x and y only through the difference $y-x$. Plugging (3.4) into (3.3), we find

$$\mathbf{E}\,B_n = 2\sum_{k=2}^{n}(-1)^k\binom{n}{k}\int_0^1\int_x^1 b(x,y)(y-x)^{k-2}\,dy\,dx.$$

But, by routine (if somewhat lengthy) calculation,

$$\begin{aligned}\int_0^1\int_x^1 b(x,y)(y-x)^{k-2}\,dy\,dx &= \sum_{\ell=0}^{\infty}(\ell+1)\iint_{0<x<y<1:\,b(x,y)=\ell+1}(y-x)^{k-2}\,dx\,dy\\ &= \sum_{\ell=0}^{\infty}(\ell+1)2^{\ell}\int_0^{2^{-(\ell+1)}}\int_{2^{-(\ell+1)}}^{2^{-\ell}}(y-x)^{k-2}\,dy\,dx\\ &= \frac{1}{(k-1)k[1-2^{-(k-1)}]}.\end{aligned}$$

This now leads immediately to the desired (3.1).

4. Asymptotic mean number of bit comparisons

Formula (1.1), repeated at (3.1), is hardly suitable for numerical calculations or asymptotic treatment, due to excessive cancellations in the alternating sum. Indeed, if (say) $n = 100$, then the terms (including the factor 2, for definiteness) alternate in sign, with magnitude as large as 10^{25}, and yet $\mathbf{E}\,B_n \doteq 2295$. Fortunately, there is a standard complex-analytic technique designed for precisely our situation (alternating binomial sums), namely, *Rice's method.* We will not review the idea behind the method here, but rather refer the reader to (for example) Section 6.4 of [16]. Let

$$h(z) := \frac{2}{(z-1)z[1-2^{-(z-1)}]}$$

and let $B(z,w) := \Gamma(z)\Gamma(w)/\Gamma(z+w)$ denote the (meromorphic continuation) of the classical beta function. According to Rice's method, $\mathbf{E}\,B_n$ equals the sum of the residues of the function $B(n+1,-z)h(z)$ at

- the triple pole at $z = 1$;
- the simple poles at $z = 1 + i\beta k$, for $k \in \mathbb{Z}\setminus\{0\}$;
- the double pole at $z = 0$.

The residues are easily calculated, especially with the aid of such symbolic-manipulation software as `Mathematica` or `Maple`. Corresponding to the above list, the residues equal

- $\frac{n}{\ln 2}\left[H_{n-1}^2 - (4-\ln 2)H_{n-1} + \frac{1}{6}(6-\ln 2)^2 + H_{n-1}^{(2)}\right]$;
- $\frac{i}{\pi k(-1-i\beta k)}\Gamma(-1-i\beta k)\frac{n!}{\Gamma(n-i\beta k)}$;
- $-2(H_n + 2\ln 2 + 1)$,

where $H_n^{(r)} := \sum_{j=1}^n j^{-r}$ denotes the nth harmonic number of order r and $H_n := H_n^{(1)}$. Summing the residue contributions gives an alternative exact formula for $\mathbf{E}\,B_n$, from which the asymptotic expansion (1.2) (as well as higher-order terms) can be read off easily using standard asymptotics for $H_n^{(r)}$ and Stirling's formula; we omit the details.

This completes the proof of Theorem 1.1.

Remark 4.1. We can calculate $\mathbf{E}\,K_n$ in the same fashion (and somewhat more easily), by replacing the bit-index function b by the constant function 1. Following this approach, we obtain first the following analogue of (3.1):

$$\mathbf{E}\, K_n = 2\sum_{k=2}^{n}(-1)^k\binom{n}{k}\frac{1}{(k-1)k}.$$

Then the residue contributions using Rice's method are

- $2n(H_n - 2 - \frac{1}{n})$, at the double pole at $z = 1$;
- $2(H_n + 1)$, at the double pole at $z = 0$.

Summing the two contributions gives an alternative derivation of (2.1).

5. Poissonized model for uniform draws

As a warm-up for Section 6, we now suppose that the number of keys (throughout this section still assumed to be uniformly distributed) is Poisson with mean λ.

5.1. **Key comparisons.**

We begin with a lemma which provides both the analogue of (2.1)–(2.2) and two other facts we will need in Section 6.

Lemma 5.1. *In the setting of Theorem 1.3 with F uniform, the expected number of key comparisons is a strictly convex function of λ given by*

$$\mathbf{E}\, K(\lambda) = 2\int_0^\lambda (\lambda - y)(e^{-y} - 1 + y)y^{-2}\, dy.$$

Asymptotically, as $\lambda \to \infty$ we have

$$\mathbf{E}\, K(\lambda) = 2\lambda\ln\lambda - (4 - 2\gamma)\lambda + 2\ln\lambda + 2\gamma + 2 + O(e^{-\lambda}\lambda^{-2}) \tag{5.1}$$

and as $\lambda \to 0$ we have

$$\mathbf{E}\, K(\lambda) = \tfrac{1}{2}\lambda^2 + O(\lambda^3). \tag{5.2}$$

Comparing the $n \to \infty$ expansion (2.2) with the corresponding expansion for Poisson(λ) many keys, note the difference in constant terms and the much smaller error term in the Poisson case.

Proof. To obtain the exact formula, begin with

$$\mathbf{E}\, K_n = \int_0^1\int_x^1 p_n(x, y)\, dy\, dx;$$

cf. (3.3) and recall Remark 4.1. Then multiply both sides by $e^{-\lambda}\lambda^n/n!$ and sum, using the middle expression in (3.4); we omit the simple computation. Strict convexity then follows from the calculation $\frac{d^2}{d\lambda^2}\mathbf{E}\, K(\lambda) = 2(e^{-\lambda} - 1 + \lambda)/\lambda^2 > 0$, and asymptotics as $\lambda \to 0$ are trivial: $\mathbf{E}\, K(\lambda) = 2\int_0^\lambda(\lambda - y)[\frac{1}{2} + O(y)]\, dy = \frac{1}{2}\lambda^2 + O(\lambda^3)$.

To derive the result for $\lambda \to \infty$, letting $\mathbf{1}[A]$ denote 1 if A holds and 0 otherwise, we observe

$$\begin{aligned}
&\tfrac{1}{2}\mathbf{E}\,K(\lambda)\\
&= \lambda \int_0^\infty \bigl(e^{-y} - 1 + y\mathbf{1}[y<1]\bigr)\, y^{-2}\, dy - \lambda \int_\lambda^\infty (e^{-y}-1) y^{-2}\, dy + \lambda \int_1^\lambda y^{-1}\, dy\\
&\qquad - \int_0^\infty \bigl(e^{-y} - \mathbf{1}[y<1]\bigr)\, y^{-1}\, dy + \int_\lambda^\infty e^{-y} y^{-1}\, dy + \int_1^\lambda y^{-1}\, dy - \int_0^\lambda dy\\
&= -\lambda(1-\gamma) + \left[1 - \lambda \int_\lambda^\infty e^{-y} y^{-2}\, dy\right] + \lambda \ln \lambda\\
&\qquad + \gamma + \int_\lambda^\infty e^{-y} y^{-1}\, dy + \ln\lambda - \lambda\\
&= \lambda \ln \lambda - (2-\gamma)\lambda + \ln\lambda + \gamma + 1 + O(e^{-\lambda}\lambda^{-2}),
\end{aligned}$$

as desired. The calculations

$$\int_0^\infty \bigl(e^{-y} - \mathbf{1}[y<1]\bigr)\, y^{-1}\, dy = -\gamma, \tag{5.3}$$

$$\int_0^\infty \bigl(e^{-y} - 1 + y\mathbf{1}[y<1]\bigr)\, y^{-2}\, dy = -(1-\gamma), \tag{5.4}$$

$$\int_\lambda^\infty e^{-y} y^{-1}\, dy = e^{-\lambda}\lambda^{-1} + O(e^{-\lambda}\lambda^{-2}), \tag{5.5}$$

$$\int_\lambda^\infty e^{-y} y^{-2}\, dy = e^{-\lambda}\lambda^{-2} + O(e^{-\lambda}\lambda^{-3}), \tag{5.6}$$

used at the second and third equalities are justified in Appendix A. □

Remark 5.2. The error term in (5.1) can, using Lemma A.2, be refined to an asymptotic expansion. Indeed, for any $M \geq 1$ it can be written as

$$e^{-\lambda} \sum_{k=1}^{M-1} (-1)^{k+1} k \cdot k!\, \lambda^{-k-1} + O(e^{-\lambda}\lambda^{-M-1}).$$

To handle the number of bit comparisons, we will also need the following bounds on the moments of $K(\lambda)$. Together with Lemma 5.1, these bounds also establish concentration of $K(\lambda)$ about its mean when λ is large. For real $1 \leq p < \infty$, we let $\|W\|_p := (\mathbf{E}\,|W|^p)^{1/p}$ denote L^p-norm and use $\mathbf{E}(W;A)$ as shorthand for the expectation of the product of W and the indicator of the event A.

Lemma 5.3. *For every real $p \geq 1$, there exists a constant $c_p < \infty$ such that*

$$\begin{aligned}
&\|K(\lambda) - \mathbf{E}\,K(\lambda)\|_p \leq c_p \lambda && \text{for } \lambda \geq 1,\\
&\|K(\lambda)\|_p \leq c_p \lambda^{2/p} && \text{for } \lambda \leq 1.
\end{aligned}$$

In particular, $\mathbf{Var} K(\lambda) \leq c_2^2 \lambda^2$ *for all* $\lambda > 0$.

Proof. We use the notation of Theorem 1.3 with F uniform [so that $K(\lambda) = K_N$ with N distributed Poisson(λ)] and write $\kappa_n := \mathbf{E}\,K_n$ for $n \geq 0$.

(a) The first result is certainly true for $\lambda \geq 1$ bounded away from ∞. For $\lambda \to \infty$ the result can be established by Poissonizing standard `Quicksort` moment calculations, as we now sketch. (Although the following argument is valid for all $p \geq 1$, the reader that so prefers may assume that p is an even integer.) We start with

$$\|K(\lambda) - \mathbf{E}\,K(\lambda)\|_p \le \|K_N - \kappa_N\|_p + \|\kappa_N - \mathbf{E}\,K(\lambda)\|_p \tag{5.7}$$

and proceed to argue that the first term on the right is asymptotically linear in λ while the second term is $o(\lambda)$.

To handle the first term, observe that

$$\|K_N - \kappa_N\|_p^p = \mathbf{E}|K_N - \kappa_N|^p = \mathbf{E}\,\mathbf{E}[|K_N - \kappa_N|^p \,|\, N].$$

But

$$\mathbf{E}[|K_N - \kappa_N|^p \,|\, N = n] = \mathbf{E}|K_n - \kappa_n|^p;$$

by the comments at the very end of Section 2 this equals $(1 + o(1)) \left(\mathbf{E}\,|\widehat{K}|^p\right) n^p$ as $n \to \infty$ and so can be bounded for all n by a constant times n^p. Thus one need only observe that $\mathbf{E}\,N^p = (1 + o(1))\lambda^p$ as $\lambda \to \infty$ to complete treatment of the first term on the right in (5.7).

To treat the second term in RHS(5.7) as $\lambda \to \infty$, one can show using (2.2) and (5.1) and the normal approximation to the Poisson that

$$\|\kappa_N - \mathbf{E}\,K(\lambda)\|_p = (1 + o(1))\,2\|N \ln N - \lambda \ln \lambda\|_p = (1 + o(1))2\|Z\|_p\,\lambda^{1/2} \ln \lambda = o(\lambda)$$

where Z has the standard normal distribution. We omit the details.

(b) For $\lambda \le 1$ we use

$$\mathbf{E}\,K^p(\lambda) \le \mathbf{E}\left[\binom{N}{2}^p; N \ge 2\right] \le \mathbf{E}\,[N^{2p}; N \ge 2] = \lambda^2 \sum_{n=2}^{\infty} e^{-\lambda} \frac{\lambda^{n-2}}{n!} n^{2p} \le c_p^p \lambda^2,$$

provided c_p is taken to be at least the finite value $\left[\sum_{n=2}^{\infty} (n^{2p}/n!)\right]^{1/p}$. □

5.2. **Bit comparisons.** We now turn our attention from $K(\lambda)$ to the more interesting random variable $B(\lambda)$, the total number of bit comparisons. We discuss first asymptotics for the mean $\mu_{\text{unif}}(\lambda)$ and then the variability of $B(\lambda)$ about the mean. In our next proposition we will derive the asymptotic estimate (1.5) by applying standard asymptotic techniques to the exact formula (1.4).

Proposition 5.4. *Asymptotically as $\lambda \to \infty$, we have*

$$\mu_{\text{unif}}(\lambda) = \mathbf{E}\,B(\lambda) = \lambda(\ln \lambda)(\lg \lambda) - c_1 \lambda \ln \lambda + c_2 \lambda + \pi_\lambda \lambda + O(\log \lambda).$$

Proof (outline). Recalling (1.4) and noting that for $x > 0$ we have

$$\sum_{k=2}^{\infty} (-1)^k \frac{x^k}{k!(k-1)k} = \int_0^x \int_0^w v^{-2}(e^{-v} - 1 + v)\,dv\,dw =: g(x),$$

it follows that $\mu(\lambda) \equiv \mu_{\text{unif}}(\lambda)$ has the harmonic sum form

$$\mu(\lambda) = 2 \sum_{j=0}^{\infty} 2^j g(2^{-j}\lambda),$$

rendering it amenable to treatment by Mellin transforms, see, e.g., [10] or [11]. Indeed, it follows immediately that the Mellin transform μ^* of μ is given for s in the fundamental strip $\{s \in \mathbb{C} : -2 < \operatorname{Re} s < -1\}$ by

$$\mu^*(s) = 2g^*(s)\Lambda(s)$$

in terms of the Mellin transform g^* of g and the generalized Dirichlet series

$$\Lambda(s) = \sum_{j=0}^{\infty} 2^{j(s+1)} = \frac{1}{1 - 2^{s+1}}.$$

But it's also easy to check using the integral formula for g that

$$g^*(s) = \frac{\Gamma(s)}{(s+1)s},$$

and so

$$\mu^*(s) = \frac{2\Gamma(s)}{(s+1)s(1-2^{s+1})}.$$

The desired asymptotic expansion for $\mu(\lambda)$ (including the remainder term) can then be read off from the singular behavior of $\mu^*(s)$ at its poles located at $s = -1$ (triple pole), $s = -1 - i\beta k$ for $k \in \mathbb{Z} \setminus \{0\}$ (simple poles), and $s = 0$ (double pole), paralleling the use of Rice's method for $\mathbf{E}\, B_n$ in Section 4. □

In order to move beyond the mean of $B(\lambda)$, we define

$$I_{k,j} := [(j-1)2^{-k}, j2^{-k})$$

to be the jth dyadic rational interval of rank k, and consider

$$B_k(\lambda) := \text{number of comparisons of } (k+1)\text{st bits},$$
$$B_{k,j}(\lambda) := \text{number of comparisons of } (k+1)\text{st bits between keys in } I_{k,j}.$$

Observe that

$$B(\lambda) = \sum_{k=0}^{\infty} B_k(\lambda) = \sum_{k=0}^{\infty} \sum_{j=1}^{2^k} B_{k,j}(\lambda). \tag{5.8}$$

A simplification provided by our Poissonization is that, for each fixed k, the variables $B_{k,j}(\lambda)$ are independent. Further, the marginal distribution of $B_{k,j}(\lambda)$ is simply that of $K(2^{-k}\lambda)$.

Remark 5.5. Taking expectations in (5.8), we find

$$\mu_{\text{unif}}(\lambda) = \mathbf{E}\, B(\lambda) = \sum_{k=0}^{\infty} 2^k\, \mathbf{E}\, K(2^{-k}\lambda). \tag{5.9}$$

If one is satisfied with a remainder of $O(\lambda)$ rather than $O(\log \lambda)$, then Proposition 5.4 can also be proved by means of (5.9). This is done by splitting the sum $\sum_{k=0}^{\infty}$ there into $\sum_{k=0}^{\lfloor \lg \lambda \rfloor}$ and $\sum_{k=\lfloor \lg \lambda \rfloor + 1}^{\infty}$ and utilizing (5.1) (to the needed order) for the first sum and (5.2) [or rather the simpler $\mathbf{E}\, K(\lambda) = O(\lambda^2)$ as $\lambda \to 0$] for the second. We omit the details. (See also Section 6 where this argument is used in a more general situation as part of the proof of Theorem 1.3.)

Moreover, we are now in position to establish the concentration of $B(\lambda)$ about $\mu_{\text{unif}}(\lambda)$ promised just prior to (1.6).

Proposition 5.6. *There exists a constant c such that $\mathbf{Var}\, B(\lambda) \le c^2\lambda^2$ for $0 < \lambda < \infty$.*

Proof. For $0 < \lambda < \infty$, we have by (5.8), the triangle inequality for $\|\cdot\|_2$, independence and $B_{k,j}(\lambda) \overset{\mathcal{L}}{=} K(2^{-k}\lambda)$, and Lemma 5.3, with $c := c_2 \sum_{k=0}^{\infty} 2^{-k/2}$,

$$[\mathbf{Var} B(\lambda)]^{1/2} \le \sum_{k=0}^{\infty} [\mathbf{Var}\, B_k(\lambda)]^{1/2} \le \sum_{k=0}^{\infty} [2^k \mathbf{Var}\, K(2^{-k}\lambda)]^{1/2} \le c\lambda. \qquad \square$$

Our next proposition extends the previous one but is limited to $\lambda \ge 1$.

Proposition 5.7. *For any real $1 \le p < \infty$, there exists a constant $c'_p < \infty$ such that*

$$\|B(\lambda) - \mathbf{E}\, B(\lambda)\|_p \le c'_p \lambda \qquad \text{for } \lambda \ge 1.$$

Proof. Because L^p-norm is nondecreasing in p, we may assume that $p \ge 2$. The proof again starts with use of the triangle inequality for $\|\cdot\|_p$: For $0 < \lambda < \infty$ we have from (5.8) that

$$\|B(\lambda) - \mathbf{E}\, B(\lambda)\|_p \le \sum_{k=0}^{\infty} \|B_k(\lambda) - \mathbf{E}\, B_k(\lambda)\|_p. \tag{5.10}$$

Further,

$$B_k(\lambda) - \mathbf{E} B_k(\lambda) = \sum_{j=1}^{2^k} \bigl[B_{k,j}(\lambda) - \mathbf{E} B_{k,j}(\lambda)\bigr],$$

where the summands are independent and centered, each with the same distribution as $K(2^{-k}\lambda) - \mathbf{E} K(2^{-k}\lambda)$. Hence, by Rosenthal's inequality [19, Theorem 3] (see also, e.g., [12, Theorem 3.9.1]) and Lemma 5.3,

$$\begin{aligned}
\|B_k(\lambda) - \mathbf{E} B_k(\lambda)\|_p &\le b_1 \Bigl(2^{k/p} \|B_{k,j}(\lambda) - \mathbf{E} B_{k,j}(\lambda)\|_p + 2^{k/2} \|B_{k,j}(\lambda) - \mathbf{E} B_{k,j}(\lambda)\|_2\Bigr) \\
&= b_1 2^{k/p} \|K(2^{-k}\lambda) - \mathbf{E} K(2^{-k}\lambda)\|_p + b_1 2^{k/2} \|K(2^{-k}\lambda) - \mathbf{E} K(2^{-k}\lambda)\|_2 \\
&\le b_1 2^{k/p} c_p \bigl(2(2^{-k}\lambda)^{2/p} + 2^{-k}\lambda\bigr) + b_1 2^{k/2} c_2 2^{-k}\lambda \\
&\le b_2 2^{-k/p} \lambda^{2/p} + b_3 2^{-k/2}\lambda
\end{aligned}$$

for some constants b_1, b_2 and b_3 (depending on p). Therefore, by (5.10),

$$\|B(\lambda) - \mathbf{E} B(\lambda)\|_p \le b'_2 \lambda^{2/p} + b'_3 \lambda \le (b'_2 + b'_3)\lambda$$

when $\lambda \ge 1$. $\square$

Remark 5.8. For the (rather uninteresting) case $\lambda \le 1$, the same proof yields $\|B(\lambda) - \mathbf{E} B(\lambda)\|_p \le c'_p \lambda^{2/p}$ for $p \ge 2$. This inequality actually holds (for some c'_p) for all $p \ge 1$; the case $1 \le p < 2$ follows easily from (5.8) and Lemma 5.3.

Remark 5.9. In [1] it is shown (in a more general setting) that the variables $B_k(\lambda)$ are positively correlated, from which it is easy to check that $\mathbf{Var}\, B(\lambda) = \Omega(\lambda^2)$ for $\lambda \ge 1$. We then have $\|B(\lambda) - \mathbf{E}\, B(\lambda)\|_p = \Theta(\lambda)$ for each real $2 \le p < \infty$. In fact, it is even true that $[B(\lambda) - \mathbf{E}\, B(\lambda)]/\lambda$ has a nondegenerate limiting distribution: see [4].

6. Mean number of bit comparisons for keys drawn from an arbitrary density f

In this section we outline martingale arguments for proving Theorem 1.3 for the expected number of bit comparisons for Poisson(λ) draws from a rather general density f. (For background on martingales, see any standard measure-theoretic probability text, e.g., [2].) In addition to the notation above, we will use the following:

$$\begin{aligned} p_{k,j} &:= \int_{I_{k,j}} f, \\ f_{k,j} &:= (\text{average value of } f \text{ over } I_{k,j}) = 2^k p_{k,j}, \\ f_k(x) &:= f_{k,j} \quad \text{for all } x \in I_{k,j}, \\ f^*(\cdot) &:= \sup_k f_k(\cdot). \end{aligned}$$

Note for each $k \geq 0$ that $\sum_j p_{k,j} = 1$ and that $f_k : (0,1) \to [0,\infty)$ is the smoothing of f to the rank-k dyadic rational intervals. From basic martingale theory we have immediately the following simple but key observation.

Lemma 6.1. *With $f_\infty := f$,*

$$(f_k)_{0 \leq k \leq \infty} \textit{ is a Doob's martingale,}$$

and $f_k \to f$ almost surely (and in L^1).

Our proof of Theorem 1.3 will also utilize the following technical lemma.

Lemma 6.2. *If (as assumed in Theorem 1.3) the probability density f on $(0,1)$ satisfies $\int_0^1 f(\ln_+ f)^4 < \infty$, then*

$$\int_0^1 f^*(\ln_+ f^*)^3 < \infty. \tag{6.1}$$

Proof. This follows readily by applying one of the standard maximal inequalities for nonnegative submartingales which asserts that for a nonnegative submartingale $(Y_k)_{1 \leq k < \infty}$ and $Y^* := \sup_{1 \leq k < \infty} Y_k$ we have

$$\mathbf{E}\, Y^* \leq \frac{e}{e-1}\left[1 + \sup_{1 \leq k < \infty} \mathbf{E}(Y_k \ln_+ Y_k)\right]; \tag{6.2}$$

see, e.g., [12, Theorem 10.9.4]. The process $(Y_k := f_k(\ln_+ f_k)^3)_{1 \leq k < \infty}$ is a submartingale by Lemma 6.1 and the convexity of the function $x \to x(\ln_+ x)^3$, and for every $1 \leq k < \infty$ we have

$$\int_0^1 Y_k \ln_+ Y_k \leq 4 \int_0^1 f_k(\ln_+ f_k)^4 \leq 4 \int_0^1 f(\ln_+ f)^4 < \infty,$$

so (6.2) does indeed give the desired conclusion. □

Before we begin the proof of Theorem 1.3 we remark that the asymptotic inequality $\mu_f(\lambda) \geq \mu_{\text{unif}}(\lambda)$ observed there in fact holds for every $0 < \lambda < \infty$.

Indeed,

$$\begin{aligned}\mu_f(\lambda) &= \sum_{k=0}^{\infty}\sum_{j=1}^{2^k} \mathbf{E}\,K(\lambda p_{k,j}) \\ &\geq \sum_{k=0}^{\infty} 2^k \mathbf{E}\,K(\lambda 2^{-k}) = \mu_{\mathrm{unif}}(\lambda),\end{aligned} \tag{6.3}$$

where the first equality appropriately generalizes (5.9), the inequality follows by the convexity of $\mathbf{E}\,K(\lambda)$ (recall Lemma 5.1), and the second equality follows by (5.9). Furthermore, strict inequality $\mu_f(\lambda) > \mu_{\mathrm{unif}}(\lambda)$ holds unless $p_{k,j} = 2^{-k}$ for all k and j, i.e., unless the distribution F is uniform. (This argument is valid also if F does not have a density.)

Proof of Theorem 1.3. Assume $\lambda \geq 1$ and, with $m \equiv m(\lambda) := \lceil \lg \lambda \rceil$, split the double sum in (6.3) as

$$\mu_f(\lambda) = \sum_{k=0}^{m}\sum_{j=1}^{2^k} \mathbf{E}\,K(\lambda p_{k,j}) + R(\lambda), \tag{6.4}$$

with $R(\lambda)$ a remainder term. Our first aim is to show that

$$R(\lambda) := \sum_{k=m+1}^{\infty}\sum_{j=1}^{2^k} \mathbf{E}\,K(\lambda p_{k,j}) = O(\lambda).$$

Since $\mathbf{E}\,K(\cdot)$ is nondecreasing, we have the inequality

$$\begin{aligned}\mathbf{E}\,K(\lambda p_{k,j}) &\leq \sum_{n=-\infty}^{\infty} \mathbf{E}\,K(2^{n+1})\,\mathbf{1}[2^n \leq \lambda p_{k,j} < 2^{n+1}] \\ &\leq \sum_{n=-\infty}^{\infty} 2^{-n}\,\mathbf{E}\,K(2^{n+1})\,\lambda p_{k,j}\,\mathbf{1}[\lambda p_{k,j} \geq 2^n].\end{aligned}$$

Now if $\lambda p_{k,j} \geq 2^n$, then for $x \in I_{k,j}$ we have

$$f^*(x) \geq f_k(x) = 2^k\,p_{k,j} \geq 2^k \lambda^{-1} 2^n \geq 2^{k-m+n}.$$

Hence

$$\begin{aligned}\mathbf{E}\,K(\lambda p_{k,j}) &\leq \sum_{n=-\infty}^{\infty} 2^{-n}\,\mathbf{E}\,K(2^{n+1})\,\lambda p_{k,j}\,\mathbf{1}[\lambda p_{k,j} \geq 2^n] \\ &\leq \lambda \sum_{n=-\infty}^{\infty} 2^{-n}\,\mathbf{E}\,K(2^{n+1}) \int_{I_{k,j}} f_k(x)\,\mathbf{1}[2^{k-m+n} \leq f^*(x)]\,dx\end{aligned}$$

and therefore

$$\begin{aligned}\sum_{j=1}^{2^k} \mathbf{E}\,K(\lambda p_{k,j}) &\leq \lambda \sum_{n=-\infty}^{\infty} 2^{-n}\,\mathbf{E}\,K(2^{n+1}) \int_0^1 f_k(x)\,\mathbf{1}[2^{k-m+n} \leq f^*(x)]\,dx \\ &\leq \lambda \int_0^1 f^*(x) \sum_{n=-\infty}^{\infty} 2^{-n}\,\mathbf{E}\,K(2^{n+1})\,\mathbf{1}[2^{k-m+n} \leq f^*(x)]\,dx.\end{aligned}$$

From this we conclude

$$\begin{aligned}R(\lambda) &\le \lambda \int_0^1 f^*(x) \sum_{n=-\infty}^{\infty} 2^{-n}\, \mathbf{E}\, K(2^{n+1}) \sum_{k=1}^{\infty} \mathbf{1}[2^{k+n} \le f^*(x)]\, dx \\ &= \lambda \int_0^1 f^*(x) \sum_{k=1}^{\infty} \sum_{n=-\infty}^{\nu(x,k)} 2^{-n}\, \mathbf{E}\, K(2^{n+1})\, dx,\end{aligned}$$

with $\nu(x,k) := \lfloor \lg f^*(x) \rfloor - k$. We proceed to bound the sum on n here. If $\nu \le 0$, then using the bound of (constant times λ^2) on $\mathbf{E}\, K(\lambda)$ from Lemma 5.1 we can bound the sum $\sum_{n \le \nu} 2^{-n}\, \mathbf{E}\, K(2^{n+1})$ by a constant (say, b') times 2^{ν}, while if $\nu > 0$ we can again use the estimates from Lemma 5.1 to bound, for some constants b_1, b_2, b'' the same sum by

$$b_1 + \sum_{n=1}^{\nu} 2^{-n}\, b_2\, (n+1) 2^{n+1} \le b'' \nu^2.$$

Therefore, for another constant b we have

$$\begin{aligned}\sum_{k=1}^{\infty} \sum_{n=-\infty}^{\nu(x,k)} 2^{-n}\, \mathbf{E}\, K(2^{n+1}) &\le \sum_{k=1}^{\lfloor \lg f^*(x) \rfloor - 1} b'' \nu^2(x,k) + \sum_{k=\lfloor \lg f^*(x) \rfloor}^{\infty} b' 2^{\nu(x,k)} \\ &\le \frac{b''}{3(\ln 2)^3} [\ln_+ f^*(x)]^3 + 2b' \le b \left(1 + [\ln_+ f^*(x)]^3\right).\end{aligned}$$

Using Lemma 6.2 we finally conclude

$$R(\lambda) \le b\, \lambda \int_0^1 f^* [1 + (\ln_+ f^*)^3] = O(\lambda).$$

Plugging $R(\lambda) = O(\lambda)$ and the consequence

$$\mathbf{E}\, K(x) = 2x \ln x - (4 - 2\gamma)x + O(x^{1/2}),$$

which holds uniformly in $0 \le x < \infty$, of Lemma 5.1 into (6.4), we find

$$\begin{aligned}\mu_f(\lambda) &= \sum_{k=0}^{m} \sum_{j=1}^{2^k} \Big[2\lambda p_{k,j} (\ln \lambda + \ln p_{k,j}) - (4 - 2\gamma)\lambda p_{k,j} + O\left((\lambda p_{k,j})^{1/2} \right) \Big] + O(\lambda) \\ &= \sum_{k=0}^{m} \Bigg[2\lambda \ln \lambda + 2\lambda \sum_{j=1}^{2^k} p_{k,j} \ln p_{k,j} - (4 - 2\gamma)\lambda + O\left(\lambda^{1/2} 2^{k/2} \right) \Bigg] + O(\lambda) \\ &= \mu_{\text{unif}}(\lambda) + 2\lambda \sum_{k=0}^{m} \int f_k \ln f_k + O(\lambda),\end{aligned}$$

where we have used the Cauchy–Schwarz inequality at the second equality and comparison with the uniform case ($f \equiv 1$) at the third.

But, by Lemma 6.1, (6.1), and the dominated convergence theorem,

$$\int f_k \ln f_k \longrightarrow \int f \ln f \text{ as } k \to \infty, \tag{6.5}$$

from which follows

$$\begin{aligned}\mu_f(\lambda) &= \mu_{\mathrm{unif}}(\lambda) + 2\lambda(\lg\lambda)\int f\ln f + o(\lambda\log\lambda)\\ &= \mu_{\mathrm{unif}}(\lambda) + 2\lambda(\ln\lambda)\int f\lg f + o(\lambda\log\lambda),\end{aligned}$$

as desired. □

Remark 6.3. If we make the stronger assumption that

$$f \text{ is Hölder}(\alpha) \text{ continuous on } [0,1] \text{ for some } \alpha > 0,$$

then we can quantify (6.5) and improve the $o(\lambda\log\lambda)$ remainder in the statement of Theorem 1.3 to $O(\lambda)$. A proof is provided in Appendix B.

7. An improvement: BitsQuick

Recall the operation of Quicksort described in Section 2. Suppose that the pivot [call it $x = 0.x(1)\,x(2)\,\dots$] has its first m_1 bits $x(1), x(2), \dots, x(m_1)$ all equal to 0. Then the subarray of keys smaller than x all have length-m_1 prefix consisting of all 0s as well, and it wastes time to compare these known bits when Quicksort is called recursively on this subarray.

We call BitsQuick the obvious recursive algorithm that does away with this waste. We give one possible implementation in the boxed pseudocode, which calls for some explanation. The initial call to the routine BitsQuick(A, m) is to BitsQuick$(A_0, 0)$, where A_0 is the full array to be sorted; in general, the routine BitsQuick(A, m) in essence sorts a subarray A of A_0 in which every element has (and is known to have) the same prefix of length m

There, for $m_1 = 0, 1, \dots$, we use the notation $L^{m_1}(y)$ for the result of rotating to the left m_1 bits the register containing key y—i.e., replacing $y = .y(1)\,y(2)\,\dots$ by $.y(m_1+1)\,y(m_1+2)\,\dots$. The input m indicates how many bits each element of the array A needs to be rotated to the right before the routine terminates, and $R^m(A)$ (in the last line of the pseudocode) is the resulting array after these right-rotations. The symbol $\|$ denotes concatenation (of sorted arrays). (We omit minor implementational details, such as how to do sorting in place and to maintain random ordering for the generated subarrays, that are the same as for Quicksort and very well known.) The routine BitsQuick(A, m) returns the sorted version of A.

A related but somewhat more complicated algorithm has been considered by Roura [21, Section 5].

The following theorem is the analogue for BitsQuick of Theorem 1.1.

Theorem 7.1. *If the keys $X_1, \dots, X_n$ are independent and uniformly distributed on $(0,1)$, then the number Q_n of bit comparisons required to sort these keys using* BitsQuick *has expectation given by the following exact and asymptotic expressions:*

$$\begin{aligned}\mathbf{E}\,Q_n &= \sum_{k=2}^{n}(-1)^k\binom{n}{k}k^{-1}\left[\frac{2(k-2)}{1-2^{-k}} - \frac{k-4}{1-2^{-(k-1)}}\right] + 2nH_n - 5n + 2H_n + 1\\ &= \Bigl(2 + \frac{3}{\ln 2}\Bigr)n\ln n - \tilde{c}_1 n + \tilde{\pi}_n n + O(\log^2 n),\end{aligned}$$

where, with $\beta := 2\pi/\ln 2$ as before,

$$\tilde{c}_1 := \frac{7}{\ln 2} + \frac{15}{2} - \Big(\frac{3}{\ln 2} + 2\Big)\gamma \doteq 13.9$$

and

$$\tilde{\pi}_n := \frac{1}{\ln 2} \sum_{k \in \mathbb{Z}:\, k \neq 0} \frac{3 - i\beta k}{1 + i\beta k} \Gamma(-1 - i\beta k)\, n^{i\beta k}$$

is periodic in $\lg n$ with period 1 and amplitude smaller than 2×10^{-7}.

Proof. We establish only the exact expression; the asymptotic expression can be derived from it using Rice's method, just as we outlined for $\mathbf{E}\, B_n$ in Section 4. Further, in light of the exact expression (1.1) for $\mathbf{E}\, B_n$, we need only show that the *expected savings* $\mathbf{E}\, B_n - \mathbf{E}\, Q_n$ enjoyed by `BitsQuick` relative to `Quicksort` is given

```
The routine BitsQuick$(A, m)$

If $|A| \leq 1$
  Return $A$
Else
  Set $A_- \leftarrow \emptyset$ and $A_+ \leftarrow \emptyset$
  Choose a random pivot key $x = 0.x(1)\, x(2)\, \ldots$ from $A$
  If $x(1) = 0$
    Set $m_1 \leftarrow 1$
    While $x(m_1 + 1) = 0$
      Set $m_1 \leftarrow m_1 + 1$
    For $y \in A$ with $y \neq x$
      If $y < x$
        Set $y \leftarrow L^{m_1}(y)$ and then $A_- \leftarrow A_- \cup \{y\}$
      Else
        Set $A_+ \leftarrow A_+ \cup \{y\}$
    Set $A_- \leftarrow$ BitsQuick$(A_-, m_1)$ and
        $A_+ \leftarrow$ BitsQuick$(A_+, 0)$
    Set $A \leftarrow A_- \parallel \{x\} \parallel A_+$
  Else
    While $x(m_1 + 1) = 1$
      Set $m_1 \leftarrow m_1 + 1$
    For $y \in A$ with $y \neq x$
      If $y < x$
        Set $A_- \leftarrow A_- \cup \{y\}$
      Else
        Set $y \leftarrow L^{m_1}(y)$ and then $A_+ \leftarrow A_+ \cup \{y\}$
    Set $A_- \leftarrow$ BitsQuick$(A_-, 0)$ and
        $A_+ \leftarrow$ BitsQuick$(A_+, m_1)$
    Set $A \leftarrow A_- \parallel \{x\} \parallel A_+$
  Return $R^m(A)$
```

by the expression

$$\mathbf{E}\,B_n - \mathbf{E}\,Q_n = \sum_{k=2}^{n}(-1)^k \binom{n}{k} k^{-1} \left\{ \frac{-2(k-2)}{1-2^{-k}} + \frac{(k-3)(k-2)}{(k-1)\left[1-2^{-(k-1)}\right]} \right\} - (2nH_n - 5n + 2H_n + 1). \tag{7.1}$$

We use the order-statistics notation $X_{(1)}, \dots, X_{(n)}$ from Section 3. To derive (7.1), we will compute the (random) total savings for all comparisons with $X_{(i)}$ as pivot, sum over $i = 1, \dots, n$, and take the expectation. For convenience, we may assume that the algorithm chooses a pivot also in the case of a (sub)array with exactly 1 element, although it is not compared to anything; thus every key becomes a pivot. Observe that $X_{(i)}$ is compared as pivot with keys $X_{(L)}, \dots, X_{(R)}$ (except itself) and with no others, where $L \equiv L(i)$ and $R \equiv R(i)$ with $L \le i \le R$ are the (random) values uniquely determined by the condition that $X_{(i)}$ is the first pivot chosen from among $X_{(L)}, \dots, X_{(R)}$ but not (if $L \neq 1$) the first from among $X_{(L-1)}, \dots, X_{(R)}$ nor (if $R \neq n$) the first from among $X_{(L)}, \dots, X_{(R+1)}$. Hence, $X_{(i)}$ is compared as a pivot with $R - L$ other keys. The comparisons with $X_{(i)}$ as pivot are performed with the knowledge that all the keys $X_{(L)}, \dots, X_{(R)}$ have values in the interval $(X_{(L-1)}, X_{(R+1)})$, where if $L = 1$ we interpret $X_{(0)}$ as $0 = .000\dots$ and if $R = n$ we interpret $X_{(n+1)}$ as $1 = .111\dots$. The total savings gained by this knowledge is $\sum_{j \in [L,R]:\, j \neq i} [b(X_{(L-1)}, X_{(R+1)}) - 1] = (R-L)\,[b(X_{(L-1)}, X_{(R+1)}) - 1]$, where we recall that $b(x, y)$ denotes the index of the first bit at which x and y differ.

Therefore the grand total savings is

$$\begin{aligned} B_n - Q_n &= \sum_{i=1}^{n} [R(i) - L(i)] \left[b\left(X_{(L(i)-1)}, X_{(R(i)+1)}\right) - 1 \right] \\ &= \sum_{(l,r):\, 1 \le l \le r \le n} (r-l)\,[b(X_{(l-1)}, X_{(r+1)}) - 1] \left| \{ i : (L(i), R(i)) = (l, r) \} \right|, \end{aligned}$$

and so by independence we have

$$\mathbf{E}\,B_n - \mathbf{E}\,Q_n = \sum_{(l,r):\, 1 \le l \le r \le n} (r-l)\,[\mathbf{E}\,b(X_{(l-1)}, X_{(r+1)}) - 1]\,\mathbf{E}\left| \{ i : (L(i), R(i)) = (l, r) \} \right|.$$

The second expectation on the right is easily computed:

$$\mathbf{E}\left| \{ i : (L(i), R(i)) = (l, r) \} \right| = \sum_{i=l}^{r} \mathbf{P}[(L(i), R(i)) = (l, r)] = (r - l + 1)\theta(l, r)$$

where, abbreviating $r - l$ to d and writing “xor” for “exclusive or”,

$$\theta(l, r) = \begin{cases} (d+1)^{-1} - 2(d+2)^{-1} + (d+3)^{-1} & \text{if } l \neq 1 \text{ and } r \neq n \\ (d+1)^{-1} - (d+2)^{-1} & \text{if } l = 1 \text{ xor } r = n \\ (d+1)^{-1} & \text{if } l = 1 \text{ and } r = n, \end{cases}$$

so that

$$\mathbf{E}\left| \{ i : (L(i), R(i)) = (l, r) \} \right| = \begin{cases} 2[(d+2)(d+3)]^{-1} & \text{if } l \neq 1 \text{ and } r \neq n \\ (d+2)^{-1} & \text{if } l = 1 \text{ xor } r = n \\ 1 & \text{if } l = 1 \text{ and } r = n, \end{cases}$$

Therefore

$$\begin{aligned}
&\mathbf{E}\, B_n - \mathbf{E}\, Q_n \\
&= 2 \sum_{(l,r)\colon 2\le l\le r\le n-1} \frac{r-l}{(r-l+2)(r-l+3)} \left[\mathbf{E}\, b(X_{(l-1)}, X_{(r+1)}) - 1\right] \\
&\quad + \sum_{r=1}^{n-1} \frac{r-1}{r+1} \left[\mathbf{E}\, b(0, X_{(r+1)}) - 1\right] + \sum_{l=2}^{n} \frac{n-l}{n-l+2} \left[\mathbf{E}\, b(X_{(l-1)}, 1) - 1\right] \\
&\quad + (n-1)\left[\mathbf{E}\, b(0,1) - 1\right] \\
&= 2 \sum_{(l,r)\colon 2\le l\le r\le n-1} \frac{r-l}{(r-l+2)(r-l+3)} \left[\mathbf{E}\, b(X_{(l-1)}, X_{(r+1)}) - 1\right] \\
&\quad + 2 \sum_{r=1}^{n-1} \frac{r-1}{r+1} \left[\mathbf{E}\, b(0, X_{(r+1)}) - 1\right] \\
&= 2 \sum_{i=1}^{n} \sum_{j=i+2}^{n} \frac{j-i-2}{(j-i)(j-i+1)} \mathbf{E}\, b(X_{(i)}, X_{(j)}) \\
&\quad + 2 \sum_{j=2}^{n} \frac{j-2}{j} \mathbf{E}\, b(0, X_{(j)}) - q_n \\
&= 2D_n + 2E_n - q_n, \qquad (7.2)
\end{aligned}$$

where: at the second equality we have used symmetry and the observation that $b(0,1) = 1$; the last two sums are denoted D_n and E_n, respectively; and

$$\begin{aligned}
q_n &:= 2 \sum_{(l,r)\colon 2\le l<r\le n-1} \frac{r-l}{(r-l+2)(r-l+3)} + 2 \sum_{r=2}^{n-1} \frac{r-1}{r+1} \\
&= 2nH_n - 5n + 2H_n + 1. \qquad (7.3)
\end{aligned}$$

The expectation $\mathbf{E}\, b(X_{(i)}, X_{(j)})$ may be computed (for $1 \le i < j \le n$) by recalling the joint density $g_{n,i,j}$ of $(X_{(i)}, X_{(j)})$ given at (3.2). We then find

$$\begin{aligned}
\mathbf{E}\, b(X_{(i)}, X_{(j)}) &= \sum_{\ell=0}^{\infty} \mathbf{P}[b(X_{(i)}, X_{(j)}) \ge \ell + 1] \\
&= \sum_{\ell=0}^{\infty} \sum_{m=1}^{2^\ell} \iint_{(m-1)2^{-\ell}<x<y<m2^{-\ell}} g_{n,i,j}(x,y)\, dx\, dy \\
&= \sum_{\ell=0}^{\infty} \sum_{m=1}^{2^\ell} \iint_{(m-1)2^{-\ell}<x<y<m2^{-\ell}} \binom{n}{i-1, 1, j-i-1, 1, n-j} \\
&\qquad \times x^{i-1} (y-x)^{j-i-1} (1-y)^{n-j}\, dx\, dy.
\end{aligned}$$

Now, suppressing some computational details,

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=i+2}^{n}\frac{j-i-2}{(j-i)(j-i+1)}\binom{n}{i-1,1,j-i-1,1,n-j}x^{i-1}(y-x)^{j-i-1}(1-y)^{n-j}\\
&=\sum_{i=1}^{n}\sum_{j=i+2}^{n}(j-i-2)\binom{n}{i-1,j-i+1,n-j}x^{i-1}(y-x)^{j-i-1}(1-y)^{n-j}\\
&=\sum_{k=3}^{n}(k-3)\binom{n}{k}(y-x)^{k-2}\sum_{i=0}^{n-k}\binom{n-k}{i}x^{i}(1-y)^{n-k-i}\\
&=\sum_{k=3}^{n}(k-3)\binom{n}{k}(y-x)^{k-2}[1-(y-x)]^{n-k}\\
&=\frac{1}{2}\sum_{k=3}^{n}(-1)^{k}(k-3)(k-2)\binom{n}{k}(y-x)^{k-2},
\end{aligned}$$

and so

$$\begin{aligned}
D_n &= \sum_{\ell=0}^{\infty}\sum_{m=1}^{2^{\ell}}\iint_{(m-1)2^{-\ell}<x<y<m2^{-\ell}}\\
&\qquad\qquad \left[\frac{1}{2}\sum_{k=3}^{n}(-1)^{k}(k-3)(k-2)\binom{n}{k}(y-x)^{k-2}\right]dx\,dy\\
&=\frac{1}{2}\sum_{\ell=0}^{\infty}2^{\ell}\iint_{0<x<y<2^{-\ell}}\left[\sum_{k=3}^{n}(-1)^{k}(k-3)(k-2)\binom{n}{k}(y-x)^{k-2}\right]dx\,dy\\
&=\frac{1}{2}\sum_{k=3}^{n}(-1)^{k}\frac{(k-3)(k-2)}{(k-1)k}\binom{n}{k}\sum_{\ell=0}^{\infty}2^{-\ell(k-1)}\\
&=\frac{1}{2}\sum_{k=2}^{n}(-1)^{k}\binom{n}{k}k^{-1}\frac{(k-3)(k-2)}{(k-1)[1-2^{-(k-1)}]}. \qquad (7.4)
\end{aligned}$$

Similarly (and somewhat more easily), one sees (for $1 \le j \le n$) that

$$\begin{aligned}
\mathbf{E}\,b(0,X_{(j)}) &= \sum_{\ell=0}^{\infty}\mathbf{P}[b(0,X_{(j)})\ge \ell+1]\\
&=\sum_{\ell=0}^{\infty}\int_{0}^{2^{-\ell}}\binom{n}{j-1,1,n-j}y^{j-1}(1-y)^{n-j}\,dy
\end{aligned}$$

and that

$$\sum_{j=2}^{n}\frac{j-2}{j}\binom{n}{j-1,1,n-j}y^{j-1}(1-y)^{n-j}=\sum_{k=2}^{n}(-1)^{k-1}(k-2)\binom{n}{k}y^{k-1},$$

whence

$$
\begin{aligned}
E_n &= \sum_{\ell=0}^{\infty} \int_0^{2^{-\ell}} \left[\sum_{k=2}^{n} (-1)^{k-1} (k-2) \binom{n}{k} y^{k-1} \right] dy \\
&= \sum_{k=2}^{n} (-1)^{k-1} \frac{k-2}{k} \binom{n}{k} \sum_{\ell=0}^{\infty} 2^{-\ell k} \\
&= \sum_{k=2}^{n} (-1)^{k-1} \binom{n}{k} k^{-1} \frac{k-2}{1-2^{-k}}. \qquad (7.5)
\end{aligned}
$$

Plugging (7.3)–(7.5) into (7.2), we obtain (7.1), thus completing the proof. □

Appendix A. Some calculus

The following calculus lemmas establish the calculations (5.3)–(5.6) used in the proof of Lemma 5.1.

Lemma A.1. *Define*

$$
\begin{aligned}
\gamma_0(z) &:= \int_0^\infty e^{-y} y^z \, dy, && \operatorname{Re} z > -1; \\
\gamma_1(z) &:= \int_0^\infty \left(e^{-y} - \mathbf{1}[y<1]\right) y^z \, dy, && \operatorname{Re} z > -2; \\
\gamma_2(z) &:= \int_0^\infty \left(e^{-y} - 1 + y\mathbf{1}[y<1]\right) y^z \, dy, && -3 < \operatorname{Re} z < -1.
\end{aligned}
$$

Then the following identities hold for $z \neq -1$:

$$
\begin{aligned}
\gamma_0(z) &= \Gamma(z+1), \\
\gamma_1(z) &= (z+1)^{-1}[\gamma_0(z+1) - 1] = (z+1)^{-1}[\Gamma(z+2) - 1], \\
\gamma_2(z) &= (z+1)^{-1}[1 + \gamma_1(z+1)],
\end{aligned}
$$

and so $\gamma_1(-1) = \Gamma'(1) = -\gamma$ and $\gamma_2(-2) = -[1 + \gamma_1(-1)] = -(1-\gamma)$.

Proof. The identity for γ_0 is the definition of the function Γ, and the identities for γ_1 and γ_2 follow by integration by parts. Since $\gamma_1(z)$ is continuous in z for $\operatorname{Re} z > -2$, it follows from the identity for $\gamma_1(z)$ by passage to the limit that $\gamma_1(-1) = \Gamma'(1) = -\gamma$. Finally, we obtain the desired value of $\gamma_2(-2)$ simply by plugging $z = -2$ into the identity for $\gamma_2(z)$. □

Let $s^{\underline{k}}$ denote the falling factorial power $s(s-1)\cdots(s-k+1)$.

Lemma A.2. *For any fixed $s \in \mathbb{C}$ and $M = 0, 1, \dots,$ and all $\lambda \geq 1$,*

$$
\int_\lambda^\infty e^{-y} y^s \, dy = e^{-\lambda} \lambda^s \left[\sum_{k=0}^{M-1} s^{\underline{k}} \lambda^{-k} + O(\lambda^{-M}) \right].
$$

(The implicit constant depends on s and M, but not on λ.)

Proof. For $\lambda > 0$, let $I(\lambda; s) := \int_\lambda^\infty e^{-y} y^s \, dy$. If $\operatorname{Re} s \leq 0$, then

$$
|I(\lambda; s)| \leq \int_\lambda^\infty e^{-y} y^{\operatorname{Re} s} \, dy \leq \int_\lambda^\infty e^{-y} \lambda^{\operatorname{Re} s} \, dy = \lambda^{\operatorname{Re} s} e^{-\lambda},
$$

which yields the result for $\operatorname{Re} s \leq M = 0$.

Further, integration by parts yields

$$I(\lambda; s) = e^{-\lambda}\lambda^s + sI(\lambda; s-1),$$

and the result for $\operatorname{Re} s \le M$ follows by induction on M. Finally, if $\operatorname{Re} s > M$, we use the result just proven with M replaced by some $M' \ge \operatorname{Re} s$. □

Appendix B. Proof of Remark 6.3

We prove that if

$$f \text{ is Hölder}(\alpha) \text{ continuous on } [0,1] \text{ for some } \alpha > 0, \tag{B.1}$$

then, as claimed in Remark 6.3, the conclusion of Theorem 1.3 holds with the remainder $o(\lambda \log \lambda)$ improved to $O(\lambda)$.

Proof. Using the notation $m \equiv m(\lambda) := \lceil \lg \lambda \rceil$ of the proof of Theorem 1.3 appearing in Section 6, it follows from that proof that we need only establish the asymptotic estimate

$$\sum_{k=0}^{m} \left(\int f_k \ln f_k - \int f \ln f \right) = O(1)$$

as $\lambda \to \infty$, and for this it is clearly sufficient to show that

$$f_k(x) \ln f_k(x) - f(x) \ln f(x) = O((k+1)2^{-k\alpha}) \text{ uniformly in } x \in [0,1]. \tag{B.2}$$

But indeed (B.1) evidently implies

$$f_k(x) - f(x) = O(2^{-k\alpha}) \text{ uniformly in } x \in [0,1],$$

and thence, routinely, (B.2). □

References

[1] Patrick Bindjeme and James Allen Fill. The limiting distribution for the number of symbol comparisons used by QuickSort is nondegenerate (extended abstract). To appear in *Proceedings of the 23rd International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms*, 2012. Available from http://www.ams.jhu.edu/~fill/.

[2] Kai Lai Chung. *A Course in Probability Theory.* Third edition, Academic Press, New York, 2001.

[3] Jack Dongarra and Francis Sullivan. Guest editors' introduction: the top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23, 2000.

[4] James Allen Fill. Distributional convergence for the number of symbol comparisons used by QuickSort. *Annals of Applied Probability*, 2012, to appear. Available from http://www.ams.jhu.edu/~fill/.

[5] James Allen Fill and Svante Janson. Smoothness and decay properties of the limiting Quicksort density function. *Mathematics and Computer Science (Versailles, 2000)*, pp. 53–64. Trends Math., Birkhäuser, Basel, 2000.

[6] James Allen Fill and Svante Janson. Quicksort asymptotics. *J. Algorithms*, 44(1):4–28, 2002.

[7] James Allen Fill and Svante Janson. The number of bit comparisons used by Quicksort: an average-case analysis. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 300–307 (electronic), New York, 2004. ACM.

[8] James Allen Fill and Takehiko Nakama. Analysis of the expected number of bit comparisons required by Quickselect. *Algorithmica*, 58(3):730–769, 2010.

[9] James Allen Fill and Takehiko Nakama. Distributional convergence for the number of symbol comparisons used by QuickSelect. Preprint, 2012. Available from http://www.ams.jhu.edu/~fill/.

[10] Philippe Flajolet, Xavier Gourdon, and Philippe Dumas. Mellin transforms and asymptotics: harmonic sums. *Theor. Computer Science* 144:3–58, 1995.

[11] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*, Cambridge Univ. Press, Cambridge, UK, 2009.

[12] Allan Gut. *Probability: A Graduate Course.* Springer, New York, 2005.

[13] Charles Anthony Richard Hoare. Quicksort. *Comput. J.*, 5:10–15, 1962.

[14] Charles Knessl and Wojciech Szpankowski. Quicksort algorithm again revisited. *Discrete Math. Theor. Comput. Sci.*, 3(2):43–63 (electronic), 1999.

[15] Donald Ervin Knuth. *The Art of Computer Programming. Vol. 3: Sorting and Searching.* 2nd ed., Addison-Wesley, Reading, Mass., 1998.

[16] Hosam M. Mahmoud. *Evolution of Random Search Trees.* John Wiley & Sons Inc., New York, 1992.

[17] Ralph Neininger and Ludger Rüschendorf. Rates of convergence for Quicksort. *J. Algorithms* 44(1):51–62, 2002.

[18] Mireille Régnier. A limiting distribution for quicksort. *RAIRO Inform. Théor. Appl.*, 23(3):335–343, 1989.

[19] Haskell P. Rosenthal. On the subspaces of L^p ($p > 2$) spanned by sequences of independent random variables. *Israel J. Math.*, 8:273–303, 1970.

[20] Uwe Rösler. A limit theorem for "Quicksort". *RAIRO Inform. Théor. Appl.*, 25(1):85–100, 1991.

[21] Salvador Roura. Digital access to comparison-based tree data structures and algorithms. *J. Algorithms*, 40(1):1–23, 2001.

[22] Brigitte Vallée, Julian Clément, James Allen Fill, and Philippe Flajolet. The number of symbol comparisons in Quicksort and Quickselect. In S. Albers et al., editor, *36th International Colloquium on Automata, Languages and Programming (ICALP 2009), Part I, LNCS 5555*, pages 750–763. Springer–Verlag, 2009.

Department of Applied Mathematics and Statistics, The Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218-2682 USA
E-mail address: jimfill@jhu.edu
URL: http://www.ams.jhu.edu/∼fill/

Department of Mathematics, Uppsala University, P. O. Box 480, SE-751 06 Uppsala, Sweden
E-mail address: svante.janson@math.uu.se
URL: http://www.math.uu.se/∼svante/