Acta Informatica 31, 673–696 (1994)

Acta Informatica

Mellin transforms and asymptotics

The mergesort recurrence

Philippe Flajolet [1] **and Mordecai Golin** [2]

[1] Algorithms Project, INRIA Rocquencourt, F-78153 Le Chesnay Cedex, France
[2] Department of Computer Science, University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Received January 7, 1993 / December 14, 1993

Abstract. Mellin transforms and Dirichlet series are useful in quantifying periodicity phenomena present in recursive divide-and-conquer algorithms. This note illustrates the techniques by providing a precise analysis of the standard top-down recursive mergesort algorithm, in the average case, as well as in the worst and best cases. It also derives the variance and shows that the cost of mergesort has a Gaussian limiting distribution. The approach is applicable to a number of divide-and-conquer recurrences.

Many algorithms are based on a recursive *divide-and-conquer* strategy of splitting a problem into two subproblems of equal or almost equal size, separately solving the subproblems, and then knitting their solutions together to find the solution to the original problem. Accordingly, their complexity is expressed by recurrences of the usual divide-and-conquer form

$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n,$$

where the initial condition, f_1, and the "knitting costs", e_n, depend on the problem being studied. Typical examples are mergesort, heapsort, Karatsuba's multiprecision multiplication, discrete Fourier transforms, binomial queues, sorting networks, etc. It is relatively easy to determine general orders of growth for solutions to these recurrences as explained in standard texts, see the "master theorem" of [6, p. 62]. However, a precise asymptotic analysis is often appreciably more delicate.

At a more detailed level, divide-and-conquer recurrences tend to have solutions that involve periodicities, many of which are of a *fractal* nature. It is our purpose here to discuss the analysis of such periodicity phenomena while focussing on the analysis of the standard top-down recursive mergesort algorithm. For example, as we shall soon see, the average cost of running mergesort on n keys satisfies

$$U(n) = n \lg n + nB(\lg n) + o(n),$$

```
Algorithm MergeSort(a[1 ... n]);
if (n > 1) then
  { MergeSort (a[1 .. ⌊n/2⌋]);
    MergeSort (a[⌊n/2⌋ + 1 ..n]);
    Merge (a[1 ... ⌊n/2⌋] , a[⌊n/2⌋ + 1 ... n] , a[1 ... n]);}
```

Fig. 1. Top-down recursive mergesort

where $B(x)$ is a fractal-like periodic function. Similarly, the variance of the cost of mergesort is

$$V(n) = nC(\lg n) + o(n),$$

where $C(x)$ is also a fractal-like periodic function.

The methods employed – Mellin transforms, Dirichlet series, Perron's formula – borrow from classical analytic number theory [4]. Related problems with emphasis on digit sums and exact summatory formulae are discussed in [11].

In Sect. 1 we quickly review the mergesort algorithm and derive the equations that describe its behavior. In Sect. 2 we introduce the analytic tools that we will use and then utilize them to develop a general technique for deriving precise asymptotics of divide-and-conquer recurrences. In Sect. 3 we apply this general technique to quickly (re)derive the (already known) worst and best case costs of mergesort. In Sect. 4 we apply the technique to derive the average case cost of mergesort. In Sect. 5 we discuss the actual distribution of the cost of mergesort, analyzing its variance and proving that it has a Gaussian limiting distribution. We conclude the Sect. 6 by briefly sketching some other possible uses of our general technique.

In what follows we set $\lg n \equiv \log_2 n$, and use the standard notation for fractional parts, $\{u\} = u - \lfloor u \rfloor$.

A preliminary version of this paper has been presented at the 20th ICALP Conference [10].

1 Mergesort

Mergesort (Fig. 1 and see [15, p. 165] for a fuller description), sorts a file of n elements by (a) splitting it into two parts of sizes $\left\lfloor \frac{n}{2} \right\rfloor$ and $\left\lceil \frac{n}{2} \right\rceil$ respectively, (b) recursively mergesorting the two subfiles, and then (c) merging the two sorted subfiles together. The recursion terminates when $n=1$, because a file with one element is already sorted. The cost, in number of comparisons performed by mergesort, satisfies the canonical divide-and-conquer recurrence

$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n, \qquad n \geqq 2, \qquad f_1 = 0,$$

where the actual values of the e_n, the costs of the merges, depend upon whether it is the worst, best or average case that is being analyzed.

For a better understanding of the values of the e_n we require a deeper understanding of the mechanics of the merging procedure itself. Suppose A

```
Procedure Merge(A, B, D);   [Merges sorted lists A and B into list C
                             and then copies the result into list D.]
μ = size(A); ν = size(B);
while ( (μ > 0) and (ν > 0) ) do   [Compare largest elements in each list]
      if (A[μ] > B[ν])
         then {C[μ + ν] = A[μ]; μ = μ − 1;}
         else {C[μ + ν] = B[ν]; ν = ν − 1;}
[Move contents of nonempty list over to C]
S = μ + ν;
if (μ > 0)
   then for i = S downto 1 do  C[i] = A[i];
   else for i = S downto 1 do  C[i] = B[i];
for  i = 1 to μ + ν do D[i] = C[i];   [Copy C into D.]
```

Fig. 2. The merging procedure. It works by successively comparing the largest remaining elements in A and B

$=[a_1, \ldots, a_\mu]$ and $B=[b_1, \ldots, b_\nu]$ are two lists of numbers, both already sorted in nondecreasing order. The procedure *Merge*(A, B, D) as described in Fig. 2 merges the two sorted lists to form a new sorted list $C=[c_1, \ldots, c_{\mu+\nu}]$ whose elements are those of $A \cup B$ and then copies this list into list D. It does this by comparing the largest element in A to the largest element in B, removing the maximum of the two from the list in which it is located, and placing it in C. It then compares the largest remaining element in A to the largest remaining element in B and again removes the maximum, this time inserting it into the second largest spot in C. It continues this process of comparing the largest remaining elements in the two lists against each other, removing the maximum and concatenating it to the back of C, until one of the two lists is empty. It then moves all of the elements from the non-empty list over to the back of C. Since the elements remaining on the non-empty list are all smaller than the ones that have already been moved and also, are all already in sorted order, moving them over to C requires no further comparisons. In fact, if a list-based, as opposed to an array-based, merge is used, we can move all of the remaining items over to C simply by changing the address in one pointer.

The cost, in number of comparisons, of merging a size μ list with a size ν one, what we call a $\langle \mu, \nu \rangle$ merge, is $\mu+\nu-S$, where S is the number of elements left on the non-empty list at the end of the procedure.

We now proceed with the analysis of mergesort. The top-level merge performed by the algorithm is a $\left\langle \left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil \right\rangle$ one. In the worst case $S=1$ so $T(n)$, the worst case behavior of mergesort, satisfies

$$T(n)=T\left(\left\lfloor \frac{n}{2} \right\rfloor\right)+T\left(\left\lceil \frac{n}{2} \right\rceil\right)+n-1, \qquad n \geqq 2, \qquad T(1)=0.$$

The best case of a $\langle \mu, \nu \rangle$ merge occurs when all of the items in the larger file are bigger than all of the items in the smaller one and $S=\max(\mu, \nu)$. The best case of the $\left\langle \left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil \right\rangle$ merge then uses $n - \left\lceil \frac{n}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor$ comparisons so, $Y(n)$, the best case behavior of mergesort satisfies

$$Y(n) = Y\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + Y\left(\left\lceil \frac{n}{2} \right\rceil\right) + \left\lfloor \frac{n}{2} \right\rfloor, \quad n \geqq 2, \quad Y(1) = 0.$$

This occurs, for example, when the numbers $1 \dots n$ are in the file in inverted order.

The average case is much more interesting. Our derivation follows that of [14, p. 620]. To study the average case we assume that the $\mu + \nu$ elements in $A \cup B$ are the integers $1 \dots \mu + \nu$ and, further, that each of the $\binom{\mu+\nu}{\mu}$ possible partitions of the numbers into sorted lists A and B are equally likely. Recall that S is the number of items left in the nonempty list by the merging procedure; these items are the S smallest items in $A \cup B$. Thus $S \geqq s$ if and only if one of the two lists A or B contains *all* of the s smallest items and

$$\Pr(S \geqq s) = \frac{\binom{\mu+\nu-s}{\mu}}{\binom{\mu+\nu}{\mu}} + \frac{\binom{\mu+\nu-s}{\nu}}{\binom{\mu+\nu}{\mu}}. \tag{1}$$

The first summand is the probability that $1 \dots s$ are all in B, the second that they are all in A.

Summing over all s we find

$$E(S) = \sum_s \Pr(S \geqq s) = \frac{\mu}{\nu+1} + \frac{\nu}{\mu+1},$$

where $E(\,)$ is the expectation operator.

We can now analyze the average case behavior of mergesort assuming that the file $a[1 \dots n]$ contains a permutation of $1 \dots n$ and that each of the $n!$ permutations is equally likely. Because the permutations are all equally likely each partition into the sets $a\left[1 \dots \left\lfloor \frac{n}{2} \right\rfloor\right]$ and $a\left[\left\lfloor \frac{n}{2} \right\rfloor + 1 \dots n\right]$ is also equally likely to occur and therefore the analysis of the preceeding few paragraphs shows that the average number of comparisons performed by the top $\left\langle \left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil \right\rangle$ merge is $n - \gamma_n$ where

$$\gamma_n = \frac{\left\lfloor \frac{n}{2} \right\rfloor}{\left\lceil \frac{n}{2} \right\rceil + 1} + \frac{\left\lceil \frac{n}{2} \right\rceil}{\left\lfloor \frac{n}{2} \right\rfloor + 1}.$$

The fact that the original permutation was random means that the subfiles $a\left[1 \dots \left\lfloor \frac{n}{2} \right\rfloor\right]$ and $a\left[\left\lfloor \frac{n}{2} \right\rfloor + 1 \dots n\right]$ are also random permutations of the elements that they contain and we can therefore apply the same analysis as above to the subfiles. The average case number of comparisons performed by mergesort, $U(n)$, thus satisfies

$$U(n)=U\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+U\left(\left\lceil\frac{n}{2}\right\rceil\right)+n-\gamma_n, \quad n\geqq 2, \quad U(1)=0.$$

We can also derive a divide-and-conquer recurrence for the variance of mergesort. First note that the variance, δ_n of the cost of the topmost $\left\langle\left\lfloor\frac{n}{2}\right\rfloor,\left\lceil\frac{n}{2}\right\rceil\right\rangle$ merge can be calculated from (1) to be [14, ex. 5.2.4-2]

$$\delta_{2m+1}=\delta_{2m+2}=\frac{2m(m+1)^2}{(m+2)^2(m+3)}.$$

The total cost of mergesort is the sum of the costs of the $(n-1)$ individual merges performed by the algorithm. The cost of the topmost merge only depends upon the particular partition of the input set into $a\left[1\ldots\left\lfloor\frac{n}{2}\right\rfloor\right]$ and $a\left[\left\lfloor\frac{n}{2}\right\rfloor+1\ldots n\right]$. The costs of the recursive mergesorts on the two subfiles only depend upon the initial permutation of the elements in each of the two subfiles; these are independent of the actual items in the files. The cost of the topmost merge is therefore independent of the costs of all of the other merges. Also, the costs of all of the merges in the "left" subfile are independent of the costs of the merges in the "right" subfile because these costs are only functions of the internal permutations of their respective subfiles and the internal permutations of the "right" subfile is independent of that of the "left" one. Summarizing, we find that the costs of the $n-1$ merges that together compromise the mergesort are independent random variables. It follows that the variance of mergesort, the variance of the sum of the costs of the individual merges, is the sum of their variances and thus satisfies

$$V(n)=V\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+V\left(\left\lceil\frac{n}{2}\right\rceil\right)+\delta_n, \quad n\geqq 2, \quad V(1)=0.$$

To review and preview: we have shown that the worst case $T(n)$, best case $Y(n)$, and average case $U(n)$ costs of mergesort all satisfy a divide-and-conquer recurrence of the form

$$f_n=f_{\lfloor n/2\rfloor}+f_{\lceil n/2\rceil}+e_n, \tag{2}$$

where the actual values of $e_n=\Theta(n)$ depend upon the particular problem being studied. We will soon see that for each of these problems there exists a different periodic function $P(x)$ with period 1 such that

$$f_n=n\lg n+nP(\lg n)+O(1).$$

The variance $V(n)$, also satisfies an equation of the form (2) but with $e_n=O(1)$. We will see that its solution has the periodic term not in the second order asymptotics but in the leading term with $f_n=nP(\lg n)+o(n)$.

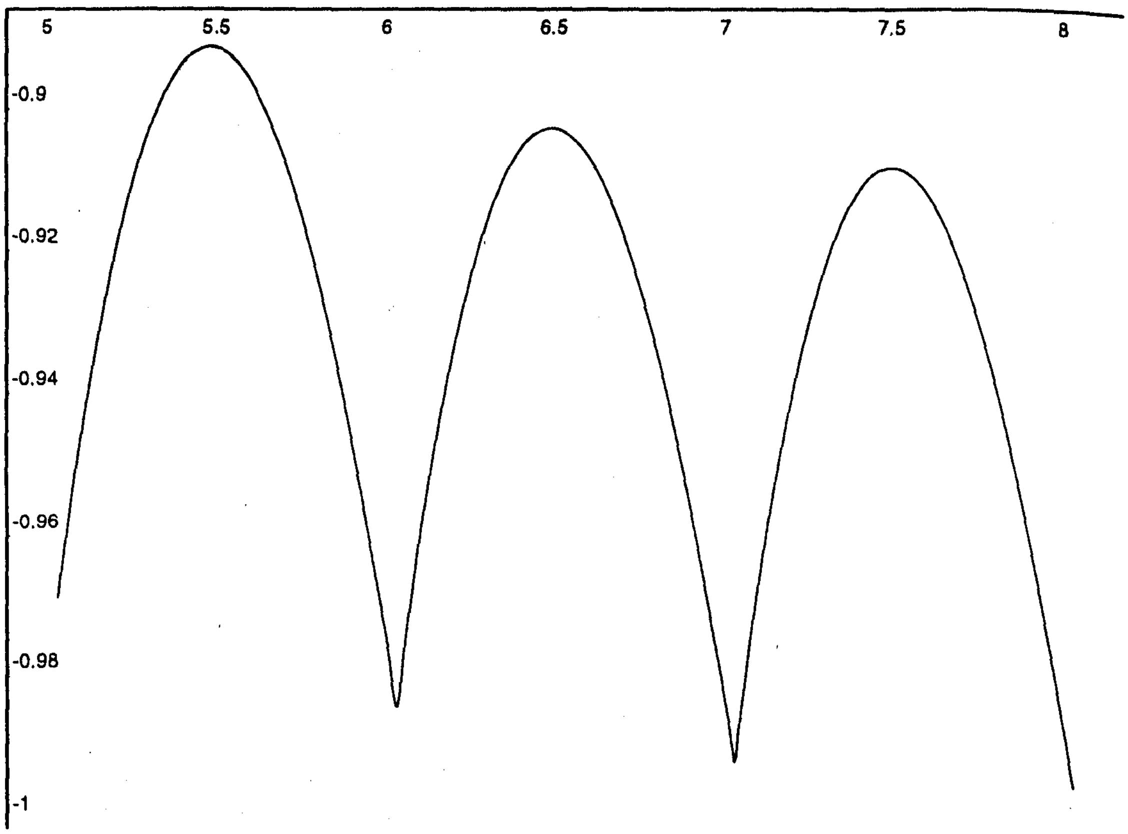

Fig. 3. The fluctuation in the worst case behavior of Mergesort, in the form of the coefficient of the linear term $\frac{1}{n}[T(n)-n\lg n]$ as a function of $\lg n \equiv \log_2 n$ for $n=32\ldots 256$. From Theorems 1 and 2, the periodic function involved, $A(u)$, fluctuates in $[-1, -0.91392]$ with mean value $a_0=-0.94269$

The periodic functions which arise in the analysis of most divide-and-conquer recurrences can be quite complicated, even fractal-like, and need special analytic tools to be studied. The case of $T(n)$, in which $e_n=n-1$, is simple enough that it can be analyzed directly. We present such a simple analysis below with the intention of giving the reader, in a setting unencumbered by heavy machinery some intuition as to where these periodic terms arise.

The precise behavior of $T(n)$ is essentially known. The main term is $n\lg n$ and $T(n)$ also contains a simple periodic function in $\lg n$. The periodicities are apparent from Fig. 3 with "cusps" whenever $\lg n$ is an integer.

Theorem 1 *The worst case cost $T(n)$ satisfies*

$$T(n)=n\lg n+nA(\lg n)+1,$$

where $A(u)$ is the periodic function

$$A(u)=1-\{u\}-2^{1-\{u\}}.$$

Proof. It is easy to check that

$$\begin{aligned}T(n)&=\sum_{k=1}^{n}\lceil \lg k\rceil\\&=n\lceil \lg n\rceil-2^{\lceil \lg n\rceil}+1.\end{aligned}$$

(See [13, p. 400], where a closely related function is discussed.) The statement then follows from writing

$$\lceil \lg n \rceil = \lg n + 1 - \{\lg n\},$$

for any n not a power of 2. □

Knuth analyzes a bottom up version of Mergesort in the average case (Algorithm L, see [14, 5.2.4 and 5.2.4-13]), when n is power of 2. Knuth's analysis is also valid for top down recursive Mergesort in this special case. When $n=2^k$, the recurrence for $U(n)$ can be unfolded to derive

$$U(2^k) = n \lg n + \beta n + o(n),$$

where

$$\beta = \sum_{j \geqq 0} \frac{1}{2^j+1} = -1.2644997803.$$

For general n, no such formula is known. (See however Eq.(17) in the proof of Theorem 4 for a related formula.) In what follows we will outline an approach that permits the analysis of mergesort type recurrences and demonstrate it by analyzing $U(n)$.

2 The mergesort recurrences

We saw in the previous section that $T(n)$, the worst case cost of mergesort, is easy to analyze because it satisfies a trivial divide-and-conquer recurrence. Most such recurrences are not as easy to attack, though. In this section we introduce a new approach to the analysis of divide-and-conquer recurrences of the mergesort type, one that works via the computation of certain associated Dirichlet series.

Let $\{w_n\}$ be a sequence of numbers. The Dirichlet generating function of w_n is defined to be

$$W(s) = \sum_{n=1}^{\infty} \frac{w_n}{n^s}.$$

The coefficients of Dirichlet series can be recovered by an inversion formula known as the Mellin-Perron formula which belongs to the galaxy of methods relating to Mellin transform analysis.

Lemma 1 (Mellin-Perron) *Assume the Dirichlet series $W(s)$ converges absolutely for $\Re(s) > 2$. Then,*

$$\text{(3)} \qquad \frac{n}{2i\pi} \int_{3-i\infty}^{3+i\infty} W(s) n^s \frac{ds}{s(s+1)} = \sum_{k=1}^{n-1} (n-k)\, w_k.$$

Proof. For completeness, we sketch the proof of this classical result. See [4] for a closely related result. For the more general version and its relation to

Mellin inversion, see for example [11] and references therein. Take $x>0$ and consider the integral

$$J(x)=\frac{1}{2i\pi}\int_{3-i\infty}^{3+i\infty} x^s \frac{ds}{s(s+1)}.$$

By closing the line of integration by a large semi-circle to the left (when $x \geqq 1$) or to the right (when $x \leqq 1$), and taking residues into account, we find that

$$J(x)=\begin{cases}0 & \text{if } x\leqq 1\\ 1-x^{-1} & \text{if } x\geqq 1\end{cases}.$$

The left hand side of Eq. (3) is therefore equal to

$$\frac{n}{2i\pi}\int_{3-i\infty}^{3+i\infty}\left(\sum_{k=1}^{\infty}\frac{w_k}{k^s}\right)n^s\frac{ds}{s(s+1)}=n\sum_{k=1}^{\infty}J\left(\frac{n}{k}\right)w_k=\sum_{k=1}^{n-1}(n-k)\,w_k$$

and the proof of the lemma follows. □

An iterated sum

$$\sum_{k=1}^{n-1}(n-k)\,w_k=\sum_{k=1}^{n-1}\sum_{l=1}^{k} w_l$$

of coefficients of a Dirichlet series is thus expressible by an integral applied to the series itself.

In order to recover the mergesort quantities $T(n)$ and $U(n)$, we determine the Dirichlet series of their second differences. Then we use the Mellin-Perron formula to derive an integral representation of the given quantity. We conclude by evaluating the integral via the residue theorem. As in other Mellin type analyses, this provides an asymptotic expansion for the quantities of interest.

This technique, which is familiar from analytic number theory, is analogous to a common technique in combinational counting. In the latter case, generating functions are ordinary, their singularities play a crucial rôle, and the asymptotic behavior of the coefficients of the power series is found by utilizing the Cauchy integral formula.

We now derive the general method for analyzing standard divide-and-conquer recurrence schemes

$$f_n=f_{\lfloor n/2\rfloor}+f_{\lceil n/2\rceil}+e_n, \tag{4}$$

for $n\geqq 2$, where e_n is a known sequence and f_n is the sequence to be analyzed. An initial condition fixing the value f_1 is also assumed. In order to make the notation unambiguous we formally set $e_0=f_0=e_1=0$. The functions $T(n)$ and $U(n)$ both satisfy this scheme: for $T(n)$, $e_n=n-1$ and for $U(n)$, $e_n=n-\gamma_n$.

Distinguishing between odd and even cases, we find that for $m>0$

$$\begin{cases}f_{2m} &=2f_m+e_{2m}\\ f_{2m+1} &=f_m+f_{m+1}+e_{2m+1}\end{cases} \tag{5}$$

Taking backward differences with $\nabla f_n = f_n - f_{n-1}$ and $\nabla e_n = e_n - e_{n-1}$ yields

$$\text{(6)} \qquad \begin{cases} \nabla f_{2m} \quad = \nabla f_m + \nabla e_{2m} \\ \nabla f_{2m+1} = \nabla f_{m+1} + \nabla e_{2m+1} \end{cases}$$

for $m > 0$. We use the symbols ∇g_m to denote $(\nabla g)_m$. For example, $\nabla f_{2m+1} = (\nabla f)_{2m+1}$. Taking forward differences of the preceding quantities, $\Delta \nabla f_n = \nabla f_{n+1} - \nabla f_n$ and $\Delta \nabla e_n = \nabla e_{n+1} - \nabla e_n$, we arrive at

$$\text{(7)} \qquad \begin{cases} \Delta \nabla f_{2m} \quad = \Delta \nabla f_m + \Delta \nabla e_{2m} \\ \Delta \nabla f_{2m+1} = \Delta \nabla e_{2m+1} \end{cases},$$

for $m \geqq 1$, with $\Delta \nabla f_1 = f_2 - 2f_1 = e_2 = \Delta \nabla e_1$. We use the symbols $\Delta \nabla g_m$ to denote $(\Delta \nabla g)_m$. For example, $\Delta \nabla f_{2m+1} = (\Delta \nabla f)_{2m+1}$.

Define the Dirichlet generating function corresponding to $w_n = \Delta \nabla f_n$,

$$W(s) = \sum_{n=1}^{\infty} \frac{\Delta \nabla f_n}{n^s}.$$

Then, multiplying w_n by n^{-s}, summing over all n, and using (7) we find

$$W(s) = \sum_{m=1}^{\infty} \frac{\Delta \nabla f_m}{(2m)^s} + \Delta \nabla f_1 + \sum_{n=2}^{\infty} \frac{\Delta \nabla e_n}{n^s} = \frac{W(s)}{2^s} + \sum_{n=1}^{\infty} \frac{\Delta \nabla e_n}{n^s}.$$

Solving for $W(s)$, we attain an explicit form for $W(s)$. Since $\sum_{k=1}^{n-1} (n-k) \Delta \nabla f_k = f_n - nf_1$ the Mellin-Perron formula yields a direct integral representation of f_n. We summarize the above discussion in:

Lemma 2 *Consider the recurrence*

$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n,$$

for $n \geqq 2$, *with* f_1 *given and* $e_n = O(n)$. *The solution satisfies*

$$\text{(8)} \qquad f_n = nf_1 + \frac{n}{2i\pi} \int_{3-i\infty}^{3+i\infty} \frac{\Xi(s) n^s}{1 - 2^{-s}} \frac{ds}{s(s+1)},$$

where

$$\Xi(s) = \sum_{n=1}^{\infty} \frac{\Delta \nabla e_n}{n^s}.$$

(The growth condition on e_n ensures existence of associated Dirichlet series when $\Re(s) > 2$, in accordance with the conditions of Lemma 1.)

Equation (8) provides an exact solution for f_n in terms of the given quantities f_1 and e_i, $i = 1, 2, \ldots$. The integral in the equation can frequently be analyzed using standard techniques yielding a precise asymptotic expression for f_n. We will now see some examples of this process.

3 Worst and best case analyses

In Sect. 1, Theorem 1, we used elementary techniques to derive an exact equation describing the worst case number of comparisons performed by mergesort. In this section we show how Lemma 2 can be utilized to rederive the same result. We do this for two reasons: the first is that it is a very clear illustration of how to apply the lemma unencumbered by analytic complications. The second is that we need this result, which expresses the cost as a Fourier series, in the next section where we derive the average case cost of the algorithm. To conclude this section we briefly sketch how to derive the best case cost of mergesort.

Theorem 2 *Let*

$$T(n)=T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+T\left(\left\lceil\frac{n}{2}\right\rceil\right)+n-1, \qquad T(1)=0$$

be the worst case cost of mergesort. Then $T(n)$ satifies

$$T(n)=n\lg n+nA(\lg n)+1$$

where $A(u)$ is a periodic function with mean value

$$a_0=\frac{1}{2}-\frac{1}{\log 2}=-0.94269\,50408,$$

and $A(u)$ has the explicit Fourier expansion,

$$A(u)=\sum_{k\in Z} a_k\, e^{2ik\pi u},$$

where, for $k\in Z\setminus\{0\}$,

$$a_k=\frac{1}{\log 2}\,\frac{1}{\chi_k(\chi_k+1)} \quad \text{with } \chi_k=\frac{2ik\pi}{\log 2}.$$

The extreme values of $A(u)$ are

$$-\frac{1+\log\log 2}{\log 2}=-0.91392, \quad \text{and} \quad -1.$$

Proof. We apply Lemma 2 with $f_n=T(n)$. For this case we have $e_n=n-1$ and $f_1=0$ so $\Delta\nabla f_1=e_2=1$ and $\Delta\nabla e_n=0$ for all $n\geqq 2$. Thus $\Xi(s)=1$ and

$$\frac{f_n}{n}=\frac{1}{2i\pi}\int_{3-i\infty}^{3+i\infty}\frac{n^s}{1-2^{-s}}\,\frac{ds}{s(s+1)}. \tag{9}$$

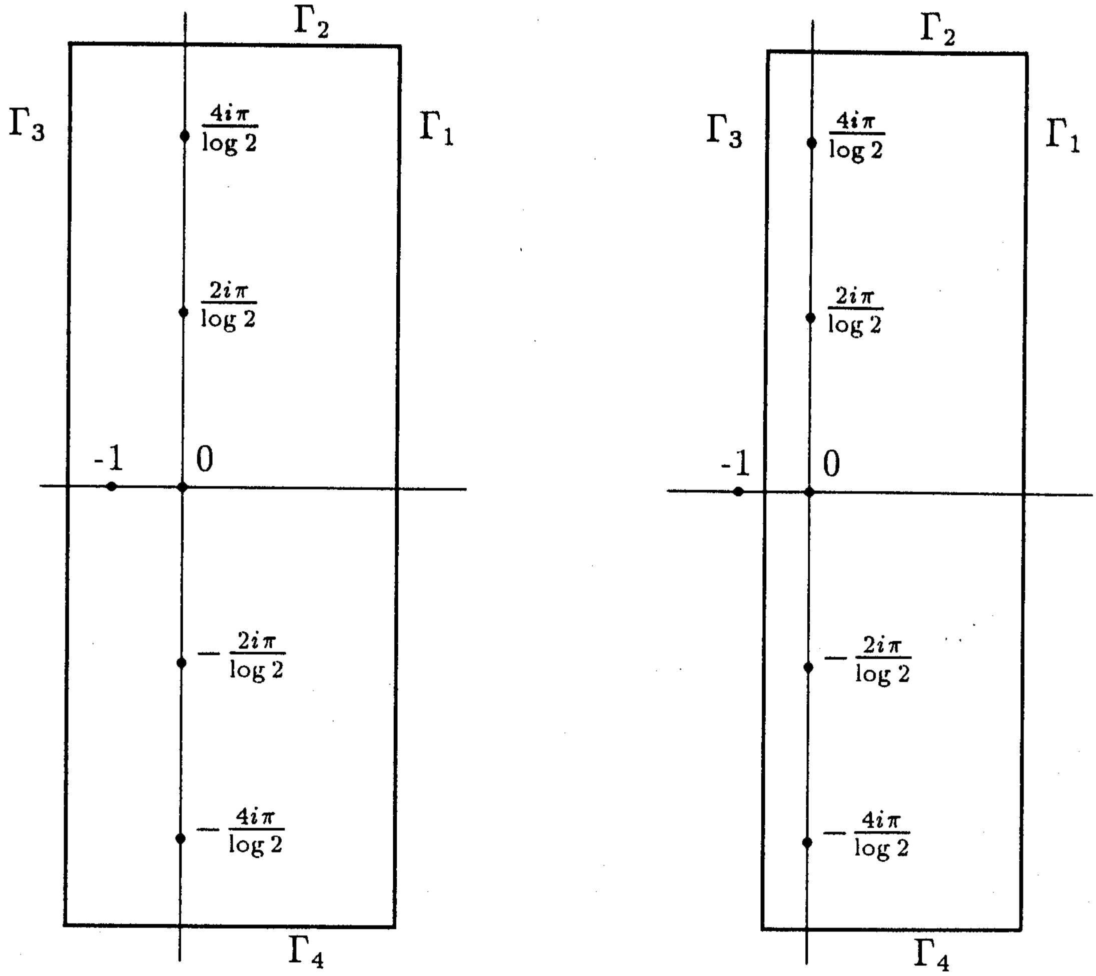

Fig. 4. The two contours employed in the proofs of Theorem 2 (left) and Theorem 3 (right). Singularities are represented by dots. Note that the contour on the left contains the singularity at -1 while the contour on the right does not

We can evaluate this integral using the residue theorem. Fix $\alpha < -1$. Let $R > 0$ and Γ be the counterclockwise contour around $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ where[1]

$$\begin{aligned}
\Gamma_1 &= \{3 + iy : |y| \leqq R\} \\
\Gamma_2 &= \{x + iR : \alpha \leqq x \leqq 3\} \\
\Gamma_3 &= \{\alpha + iy : |y| \leqq R\} \\
\Gamma_4 &= \{x - iR : \alpha \leqq x \leqq 3\}.
\end{aligned}$$

(See Fig. 4.) Set $I(s) = \frac{n^s}{1-2^{-s}} \frac{1}{s(s+1)}$ to be the kernel of the integral in (9). Letting $R \uparrow \infty$ we find that $\frac{1}{2i\pi} \int_{\Gamma_1} I(s)\,\mathrm{d}s$ becomes the integral in (9), $|\int_{\Gamma_2} I(s)\,\mathrm{d}s|$ and $|\int_{\Gamma_4} I(s)\,\mathrm{d}s|$ are both $O(1/R^2)$ and

$$\left| \int_{\Gamma_3} I(s)\,\mathrm{d}s \right| \to \left| \int_{\alpha + i\infty}^{\alpha - i\infty} I(s)\,\mathrm{d}s \right| \leqq 4n^{\alpha}.$$

[1] We further assume that R is of the form $(2j+1)\pi/\log 2$ for integer j, so that the contour passes halfway between poles of the integrand.

The residue theorem therefore yields that f_n/n equals $O(n^\alpha)$ plus the sum of the residues of $I(s)$ inside Γ.

We can actually do better. Since $I(s)$ is analytic for *all* s with $\Re(s) < -1$ we may let α go to $-\infty$ getting progressively smaller and smaller error terms. This shows that f_n/n is *exactly* equal to the sum of the residues of $I(s)$ inside Γ. The singularities of $I(s)$ are

1. A double pole at $s=0$ with residue $\lg n + \frac{1}{2} - \frac{1}{\log 2}$.
2. A simple pole at $s=-1$ with residue $\frac{1}{n}$.
3. Simple poles at $s = 2ki\pi/\log 2$, $k \in Z \setminus \{0\}$ with residues $a_k\, e^{2ik\pi \lg n}$.

Thus, as promised, we have shown that

$$T(n) = n \lg n + nA(\lg n) + 1,$$

where $A(u)$ is defined by the designated Fourier series. This Fourier series is uniformly convergent because $a_k = O(1/k^2)$.

The extreme values of $A(u)$ are calculated using standard techniques. □

We note that a computation of the Fourier series of $A(u)$ directly from Theorem 1 is also feasible and in fact yields the Fourier series derived in the last theorem (providing a convenient check on the validity of the theorem).

To analyze the best case behavior recall, from Sect. 1, that $Y(n)$, the best case number of comparisons performed by mergesort, satisfies the divide and conquer recurrence:

$$Y(n) = Y\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + Y\left(\left\lceil \frac{n}{2} \right\rceil\right) + \left\lfloor \frac{n}{2} \right\rfloor, \quad n \geqq 2, \quad Y(1) = 0. \tag{10}$$

Let $v(n)$ denote the sum of the digits of n represented in binary, for instance $v(13) = v([1101]_2) = 3$. Then by comparing recurrences, we find that

$$Y(n) = \sum_{m<n} v(m). \tag{11}$$

Equation (11) has been already noticed by several authors (see, e.g., [3]). The function $Y(n)$ has been studied by Delange [7] using elementary real analysis. It can also be subjected to the methods of this paper by applying Lemma 2 with $e_n = \lfloor n/2 \rfloor$ and $\Xi(s) = \sum_{n \geqq 1} \frac{\Delta \nabla e_n}{n^s} = \sum_{n \geqq 1} (-1)^{n+1}/n^s$. Since $Y(n)$ has been so well studied we do not go into the details of how to evaluate the integral here but, instead, refer the interested reader to [11] for a discussion of how to analyze this integral in particular and exact summatory formulae in general. The result is

Theorem 3 *The best case cost $Y(n)$ satisfies*

$$Y(n) = \tfrac{1}{2} n \lg n + nD(\lg n),$$

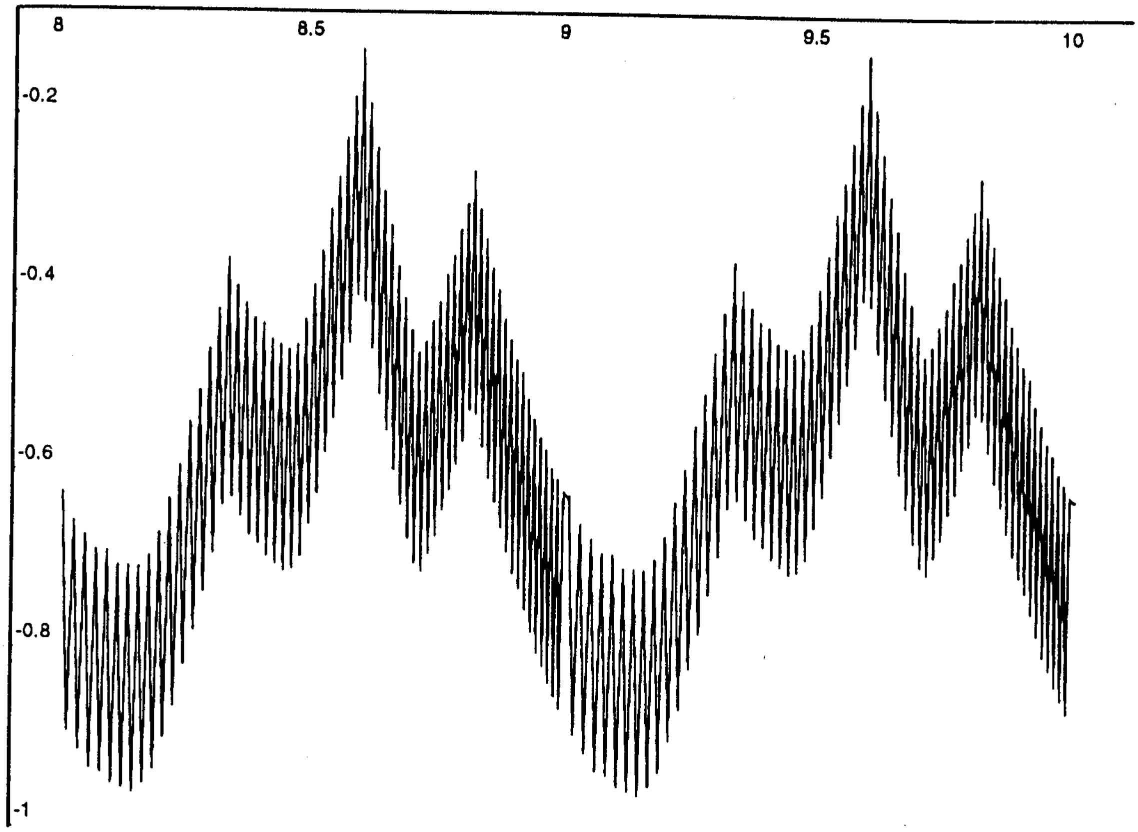

Fig. 5. The clearly fractal fluctuation in the best case behavior of mergesort, graphing the coefficient of the linear term $\frac{1}{n}\left[Y(n)-\frac{1}{2}n\lg n\right]$ using a logarithmic scale for $n=256\ldots1024$

where $D(u)$ *is a periodic function of period* 1, $D(u)=\sum_{k\in Z} d_k\, e^{2ik\pi u}$ *has Fourier coefficients*

$$d_0=\lg\sqrt{\pi}-\frac{1}{2\log 2}-\frac{1}{4},$$
$$d_k=-\frac{1}{\log 2}\,\frac{\zeta(\chi_k)}{\chi_k(\chi_k+1)},\qquad k\neq 0,\qquad \chi_k=\frac{2ik\pi}{\log 2}.$$

($\zeta(s)=\sum_{n\geq 1} 1/n^s$ *is the Riemann Zeta function.*)

Delange [7] has proven that the periodic function $D(x)$ is everywhere continuous, but that it is not differentiable at the dense set of points $\left\{\lg\left(\frac{p}{2^r}\right): p, r\in Z^+\cup\{0\}\right\}$.

4 Average case analysis

We now proceed with the major result of this paper, the analysis of the *average* number of comparisons performed by mergesort, $U(n)$.

Theorem 4 *Let* $U(n)$ *be the function which counts the average number of comparisons performed by mergesort, where*

$$U(n)=U\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+U\left(\left\lceil\frac{n}{2}\right\rceil\right)+n-\gamma_n \tag{12}$$

for $n \geqq 2$, *with* $U(1)=0$, *and*

$$\gamma_n=\frac{\left\lfloor\frac{n}{2}\right\rfloor}{\left\lfloor\frac{n}{2}\right\rfloor+1}+\frac{\left\lceil\frac{n}{2}\right\rceil}{\left\lceil\frac{n}{2}\right\rceil+1}.$$

Then the following is true:
(i). The average case cost $U(n)$ *of mergesort satisfies*

$$U(n)=n\lg n+nB(\lg n)+O(1),$$

where $B(u)$ *is a periodic function with period* 1. *Furthermore,* $B(u)$ *is everywhere continuous, but it is not differentiable at the dense set of points* $\left\{\lg\left(\frac{p}{2^r}\right): p, r\in Z^+\cup\{0\}\right\}$.
(ii). The mean value b_0 *of* $B(u)$ *equals*

$$\frac{1}{2}-\frac{1}{\log 2}-\frac{1}{\log 2}\sum_{m=1}^{\infty}\frac{2}{(m+1)(m+2)}\log\left(\frac{2m+1}{2m}\right).$$

Numerically,

$$b_0=-1.24815\,20420\,99653\,88489\ldots$$

(iii). $B(u)=\sum_{k\in Z} b_k e^{2ik\pi u}$ *where the Fourier coefficients of* $B(u)$ *are, for* $k\in Z\backslash\{0\}$,

$$b_k=\frac{1}{\log 2}\frac{1+\Psi(\chi_k)}{\chi_k(\chi_k+1)}\quad with\ \ \chi_k=\frac{2ik\pi}{\log 2},$$

and

$$\Psi(s)=\sum_{m=1}^{\infty}\frac{2}{(m+1)(m+2)}\left[\frac{-1}{(2m)^s}+\frac{1}{(2m+1)^s}\right].$$

The Fourier series is uniformly convergent to $B(u)$.
(iv). The extreme values of $B(u)$ *are*

$$\beta=-1.26449\,97803\ldots\quad and\quad -1.24075\,0572\pm 10^{-9}.$$

Proof. The proof follows the paradigm laid down by Theorem 4. We first use Lemma 2 to derive an integral form for $f_n=U(n)$ and then use residue analysis to evaluate the integral.

For $f_n=U(n)$ we are given $f_1=0$ and $\Delta\nabla f_1=e_2=1$. We are also given that for all $m>0$

$$\begin{cases} e_{2m} &=2m-2+\dfrac{2}{m+1} \\ e_{2m+1} &=2m-1+\dfrac{2}{m+2} \end{cases},\tag{13}$$

and thus

$$-\Delta\nabla e_{2m}=\frac{2}{(m+1)(m+2)}=\Delta\nabla e_{2m+1}.$$

Summing over all m we may write

$$\Xi(s)=\Delta\nabla f_1+\sum_{n=2}^{\infty}\frac{\Delta\nabla e_n}{n^s}=1+\Psi(s)$$

where

$$\Psi(s)=\sum_{m=1}^{\infty}\frac{2}{(m+1)(m+2)}\left[\frac{-1}{(2m)^s}+\frac{1}{(2m+1)^s}\right]$$

converges absolutely and is $O(1)$ on any imaginary line $\Re(s)=\alpha\geqq-1+\varepsilon$. Lemma 2 therefore tells us that

$$\frac{f_n}{n}=\frac{1}{2i\pi}\int_{3-i\infty}^{3+i\infty}\frac{n^s}{1-2^{-s}}\frac{\mathrm{d}s}{s(s+1)}+\frac{1}{2i\pi}\int_{3-i\infty}^{3+i\infty}\frac{n^s\,\Psi(s)}{1-2^{-s}}\frac{\mathrm{d}s}{s(s+1)}. \tag{14}$$

The first integral on the right-hand side was already evaluated during the proof of Theorem 2 and shown to be equal to $\lg n+A(\lg n)+\frac{1}{n}$ where $A(u)=\sum_k a_k\,e^{2ik\pi u}$.

The second integral can be evaluated using similar techniques. Let $\varepsilon>0$ and fix $\alpha=-1+\varepsilon$. Let $R>0$ and Γ be the counterclockwise contour around $\Gamma_1\cup\Gamma_2\cup\Gamma_3\cup\Gamma_4$ where

$$\Gamma_1=\{3+iy:|y|\leqq R\}$$
$$\Gamma_2=\{x+iR:\alpha\leqq x\leqq 3\}$$
$$\Gamma_3=\{\alpha+iy:|y|\leqq R\}$$
$$\Gamma_4=\{x-iR:\alpha\leqq x\leqq 3\}.$$

(See Fig. 4.) Set $I(s)=\dfrac{n^s\,\Psi(s)}{1-2^{-s}}\dfrac{1}{s(s+1)}$ to be the kernel of the second integral in (14). Letting $R\uparrow\infty$ we find that $\dfrac{1}{2i\pi}\int_{\Gamma_1} I(s)\,\mathrm{d}s$ becomes the second integral in (14), $\left|\int_{\Gamma_2} I(s)\,\mathrm{d}s\right|$ and $\left|\int_{\Gamma_4} I(s)\,\mathrm{d}s\right|$ are both $O(1/R^2)$ and

$$\left|\int_{\Gamma_3} I(s)\,\mathrm{d}s\right|\to\left|\int_{\alpha+i\infty}^{\alpha-i\infty} I(s)\,\mathrm{d}s\right|=O(n^{-1+\varepsilon}).$$

(The constants implicit in the $O(\,)$ notation are dependent upon ε.)

Thus f_n/n equals $O(n^{-1+\varepsilon})$ plus the sum of the residues calculated in Theorem 2 plus the sum of the residues of $I(s)$ inside Γ. The singularities of $I(s)$ *inside* Γ are

1. A simple pole at $s=0$ with residue $\dfrac{\Psi'(0)}{\log 2}$.
2. Simple poles at $s=\chi_k=2ki\pi/\log 2$, $k\in Z\setminus\{0\}$ with residues $b_k\, e^{2ik\pi \lg n}$.

$$\frac{1}{\log 2}\frac{\Psi(\chi_k)}{\chi_k(\chi_k+1)}\, e^{2ik\pi \lg n}=(b_k-a_k)\, e^{2ik\pi \lg n}.$$

Summing these residues, multiplying by n and then adding the previously calculated contribution from the first integral yields

$$U(n)=n\lg n+nB(\lg n)+O(n^{\varepsilon}). \tag{15}$$

Note that $I(s)$ does have a simple pole at $s=-1$ but we do not count its residue because it is outside Γ. There are technical reasons (the behavior of $\Psi(s)$ towards $\pm i\infty$ when $\Re(s)=\alpha\leqq -1$) which stop us from setting $\alpha<-1$ and surrounding this last pole by Γ. It is because $\alpha=-1+\varepsilon$ that we have the $O(n^{\varepsilon})$ error term in the expression. To refine the error estimate and go from (15) to

$$U(n)=n\lg n+nB(\lg n)+O(1),$$

we must now examine $U(n)$ in a slightly different way.

Consider the sequence $U(a2^k)$ for some fixed integer a. By unwinding the recurrence (12) we find

$$U(a2^k)=ak2^k+2^k\,U(a)-a2^k\sum_{j=0}^{k-1}\frac{1}{a2^j+1}.$$

Rewriting $U(a2^k)$ in terms of $n=a2^k$, yields, for these particular values of n,

$$U(n)=n\lg n+\beta(a)\,n+n\sum_{j=k}^{\infty}\frac{1}{a2^j+1}, \tag{16}$$

where

$$\beta(a)=\frac{U(a)}{a}-\lg a-\sum_{j=0}^{\infty}\frac{1}{a2^j+1}. \tag{17}$$

This formula generalizes the one given by Knuth [14, 5.2.4 and 5.2.4-13] for the average case when $a=1$, i.e., $n=2^k$.

B is periodic with period 1 so

$$B(\lg n)=B(\lg(a2^k))=B(k+\lg a)=B(\lg a).$$

Also, if $n=a2^k$, then $n\sum_{j=k}^{\infty}\frac{1}{a2^j+1}=O(1)$. Therefore, setting $\varepsilon<1$ in (16) and letting k go to infinity we find

$$\beta(a)=\lim_{k\to\infty}\frac{U(n)-n\lg n}{n}=B(\lg a)=B(\lg n). \tag{18}$$

Substituting back into (16) proves the sought after

$$U(n)=n\lg n+nB(\lg n)+O(1).$$

To finish the proof of the theorem note that when $\Re(s)=0$ then $|\Psi(s)|<2$, so that $b_k=O(1/k^2)$; thus the Fourier series is uniformly convergent and the function $B(u)$ is continuous. Differentiability properties and numerical estimates are discussed below. □

Non-differentiability

There is an interesting decomposition of the periodic part of the average case behavior $B(u)$ in terms of the periodic part of the worst case $A(u)$. Define first

$$A^*(u)=A(u)-a_0,\; B^*(u)=B(u)-b_0,$$

both functions having mean value 0. We have

$$B^*(u)-A^*(u)=\sum_{m=2}^{\infty}\psi_m A^*(u-\lg m), \tag{19}$$

where the ψ_m are the coefficients of the Dirichlet series $\Psi(s)=\sum_{m\geq 2}\frac{\psi_m}{m^s}$:

$$-\psi_{2m}=\frac{2}{(m+1)(m+2)}=\varphi_{2m+1}.$$

To derive (19), take the Fourier expansion of $B^*(u)-A^*(u)$, expand the Fourier coeffcients as sums since they are special values of a Dirichlet series, and exchange summations:

$$\begin{aligned}
B^*(u)-A^*(u)&=\frac{1}{\log 2}\sum_k\frac{e^{2ik\pi u}}{\chi_k(\chi_k+1)}\Psi(\chi_k)\\
&=\frac{1}{\log 2}\sum_k\frac{e^{2ik\pi u}}{\chi_k(\chi_k+1)}\sum_{m=2}^{\infty}\psi_m e^{-2ik\pi\lg m}\\
&=\frac{1}{\log 2}\sum_{m=2}^{\infty}\psi_m\left[\sum_k\frac{e^{2ik\pi(u-\lg m)}}{\chi_k(\chi_k+1)}\right]\\
&=\sum_{m=2}^{\infty}\varphi_m A^*(u-\lg m),
\end{aligned}$$

the summations on k being for $f\in Z\setminus\{0\}$.

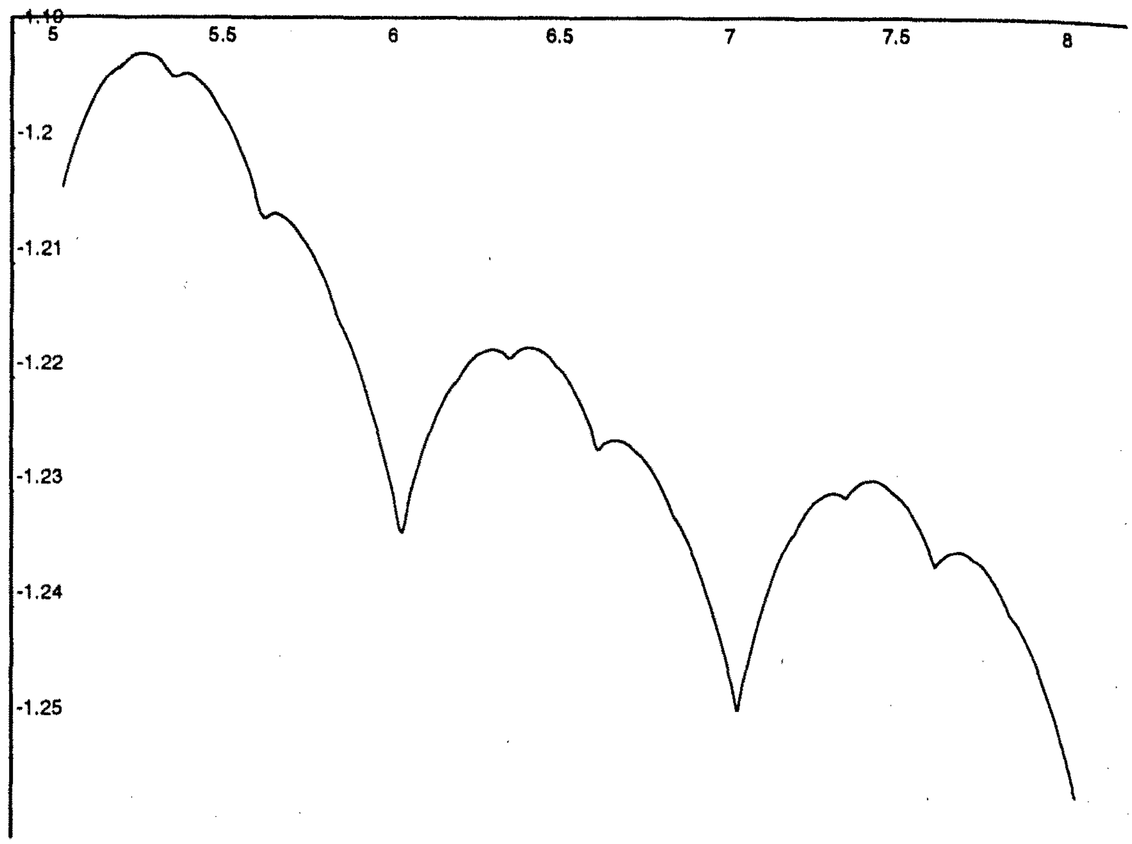

Fig. 6. The fluctuation in the average case behavior of Mergesort, graphing the coefficient of the linear term $\frac{1}{n}[U(n)-n\lg n]$ using a logarithmic scale for $n=32\ldots256$. From Theorem 3, the periodic function involved, $B(u)$, fluctuates in $[-1.26449, -1.24075]$ with mean value $b_0=-1.24815$

This unusual decomposition explains the behavior of $U(n)$ in Fig. 6. First, $A(u)$ and $A^*(u)$ have a cusp at $u=0$, where the derivative has a finite jump. The function $B^*(u)$ is $A^*(u)$ to which is added a sum of pseudo-harmonics $A^*(u-\lg m)$ with decreasing amplitudes ψ_m. The harmonics corresponding to $m=2$, 4, 8 are the same as those of $A^*(m)$ up to scaling, and their presence explains the cusp of $B^*(u)$ at $u=0$ which is visible on the graph of Fig. 6. We also have two less pronounced cusps at $\{\lg 3\}=0.58$ and at $\{\lg 5\}=0.32$ induced by the harmonics corresponding to $m=3$ and $m=5$. More generally, this decomposition allows us to prove the following property: The function $B(u)$ is *non-differentiable (cusp-like) at any point of the form* $u=\lg(p/2^r)$. Stated differently, $B(lg\ v)$ has a cusp at any dyadic rational $v=p/2^r$.

Numerical computations

These have been carried out with the help of the Maple system. The computation of the *mean value* b_0 to great accuracy can be achieved simply by appealing to a general purpose series acceleration method discussed by Vardi in his entertaining book [17]. We have $\Psi'(0)=\sum_{m=1}^{\infty}\theta(1/m)$, where

$$\theta(y)=\frac{2y^2}{(1+y)(1+2y)}\log(1+y/2).$$

The function $\theta(y)$ is analytic near $y=0$ with a singularity at $y=-1/2$. Thus

$$\theta(y)=y^3-\frac{13}{4}y^4+\frac{67}{6}y^5-\cdots=\sum_{j=3}^{\infty} c_j y^j,$$

where the $|c_j|$ grow essentially like 2^j. Select some small number m_0 (for instance $m_0=10$), and rewrite $\Psi'(0)$ as

$$\Psi'(0)=\sum_{m=1}^{m_0}\theta\left(\frac{1}{m}\right)+\sum_{j=3}^{\infty}c_j\left[\zeta(j)-\sum_{m=1}^{m_0}m^{-j}\right].$$

This form is obtained by separating the first m_0 terms, expanding each $\theta(1/m)$, and interchanging summations, which introduces the Riemann zeta function, $\zeta(s)$. Standard facts about the zeta function tell us that the infinite series converges like $(2/m_0)^j$. In this way, with 80 terms and $m_0=10$, we evaluate $\Psi'(0)$ to 50 digits in a matter of one minute of computation time.

Regarding the computation of *extreme values* of $B(u)$ accurately, the approach via the Fourier sereis does not seem to be practicable, since the Fourier coefficients only decrease as $O(k^{-2})$.

Recall instead, from (18) in the proof of the theorem, that if $n=a2^k$ then

$$B(\lg n)=B(\lg a)=\beta(a)$$

where

$$\beta(a)=\frac{U(a)}{a}-\lg a-\sum_{j=0}^{\infty}\frac{1}{a2^j+1}.$$

The computation of $\beta(a)$ for all values a in an integer interval like $[2^{15}\dots2^{16}]$ (again in a matter of minutes) then furnishes the values of B with the required accuracy.

One final note. From these estimates, Mergesort has been found to have an average case complexity about

$$n\lg n-(1.25\pm 0.01)\,n+O(1).$$

This appear to be not far from the information theoretic lower bound which is

$$\lg n!=n\lg n-n\lg e+o(n)=n\lg n-1.44n+o(n).$$

5 Variance and distribution of mergesort

In this section we derive the asymptotic behavior of the variance of top-down mergesort, prove a central limit theorem and discuss the distribution of the cost.

Recall the equation for variance, derived in Sect. 1,

$$V(n)=V\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+V\left(\left\lceil\frac{n}{2}\right\rceil\right)+\delta_n,\qquad n\geqq 2,\qquad V(1)=0,$$

where

$$\delta_{2m+1}=\delta_{2m+2}=\frac{2m(m+1)^2}{(m+2)^2(m+3)}.$$

The anaylsis unwinds exactly as in that of the average case. Applying Lemma 2 we find

Theorem 5 *The variance of the MergeSort algorithm applied to data of size n satisfies*

$$V(n)=n\cdot C(\lg n)+o(n),$$

where $C(u)$ is a continuous periodic function with period 1 and mean value

$$c_0=\frac{1}{\log 2}\sum_{m=1}^{\infty}\frac{2m(5m^2+10m+1)}{(m+1)(m+2)^2(m+3)^2}\log\frac{2m+1}{2m}$$

which evaluates to $c_0\approx 0.34549\ 95688$. $C(u)=\sum_{k\in\mathbb{Z}} c_k e^{2ik\pi u}$, where for $k\in\mathbb{Z}\setminus\{0\}$,

$$c_k=\frac{1}{\log 2}\frac{\Psi(\chi_k)}{\chi_k(\chi_k+1)}\quad \textit{with}\ \chi_k=\frac{2ik\pi}{\log 2}$$

and

$$\Psi(s)=\sum_{m=1}^{\infty}\frac{2m(5m^2+10m+1)}{(m+1)(m+2)^2(m+3)^2}\left[\frac{-1}{(2m)^s}+\frac{1}{(2m+1)^s}\right].$$

Like the function $B(u)$ that describes the fluctuation of the average cost the function $C(u)$ is continuous but non-differentiable with cusps at the logarithms of dyadic rationals, a dense set of points. Numerically, its range of fluctuation is found to lie in the interval [0.30, 0.36].

The distribution of the cost of mergesort is computable exactly, as well as numerically, using the resources of computer algebra systems. The probability generating function which describes the distribution of the number of comparisons performed by the single (topmost) merge of n elements is found from (1) to be $\xi_n(z)=$

$$\frac{2}{\binom{2m}{m}}\sum_{s=1}^{m}\binom{2m-s-1}{m-1}z^{2m-s},$$

$$\frac{1}{\binom{2m+1}{m}}\sum_{s=1}^{m+1}\left[\binom{2m-s}{m-1}+\binom{2m-s}{m}\right]z^{2m+1-s},$$

depending on whether $n=2m$ (the merge is of type $\langle m,m\rangle$) or $n=2m+1$ (the merge is of type $\langle m,m+1\rangle$). The probability generating function of the cost of merge sort then satisfies the divide-and-conquer product recurrence,

$$\Xi_n(z)=\xi_n(z)\cdot\Xi_{\lfloor n/2\rfloor}\cdot\Xi_{\lceil n/2\rceil}.$$

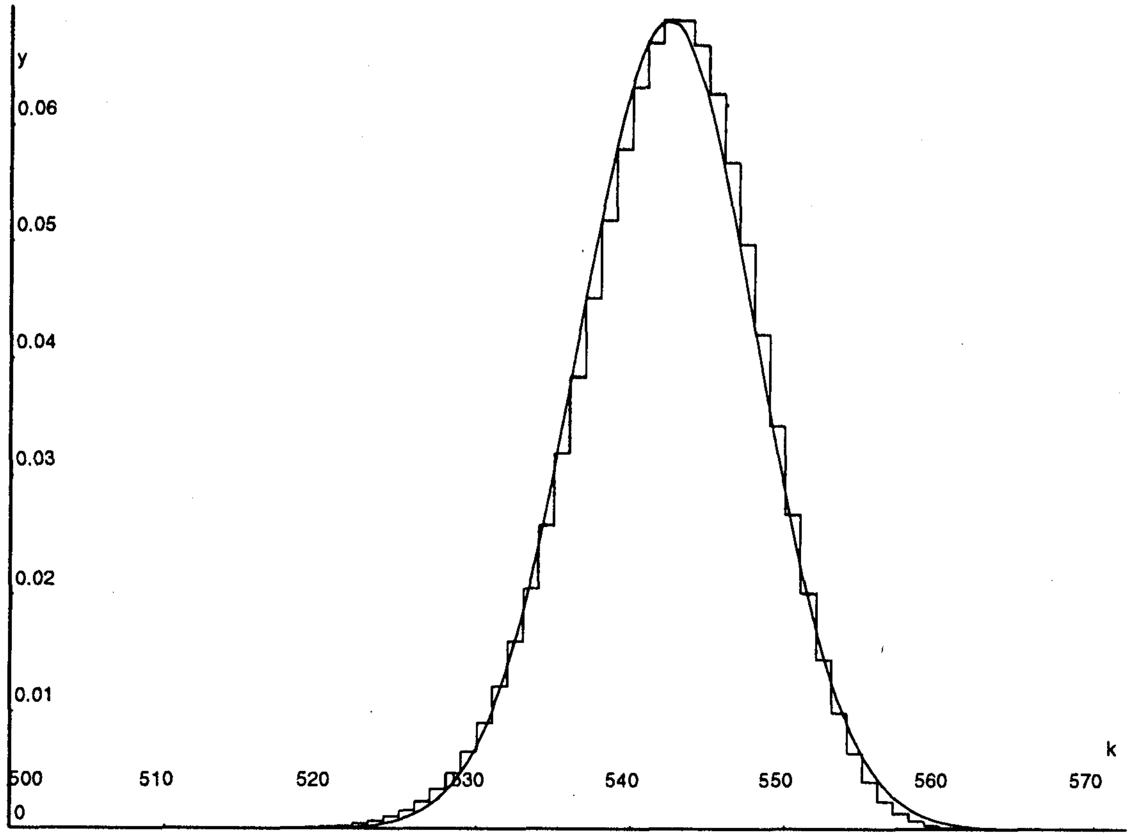

Fig. 7. The histogram of the exact probability distribution of the comparison cost of mergesort for $n = 100$

Unwinding the recurrence yields

$$\Xi_n(z) = \prod_{m \leqq m} \xi_m(z),$$

the summation being taken over the multiset of all m that appear as subfile sizes in mergesorting n elements. For instance:

$$\Xi_{23} = \xi_{23} \cdot \xi_{12} \cdot \xi_{11} \cdot \xi_6^3 \cdot \xi_5 \cdot \xi_3^7 \cdot \xi_2^8.$$

For $n = 100$, the mergesort comparison costs lie in the interval $[316 \ldots 573]$ with mean value 541.84. The standard deviation is 5.78, and Fig. 7. shows the histogram of the distribution computed from these formulae. The numerical data strongly suggest convergence to a Gaussian law with matching mean and variance that is also plotted on the same diagram.

Actually, using standard extensions of the central limit theorem to sums of independent – but not necessarily identically distributed – random variables, we find:

Theorem 6 *The cost X_n of mergesort applied to random data of size n converges in distribution to a normal variable,*

$$Pr\left\{\frac{X_n - U(n)}{\sqrt{V(n)}} \leqq \mu\right\} \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\mu} e^{-t^2/2}\, dt.$$

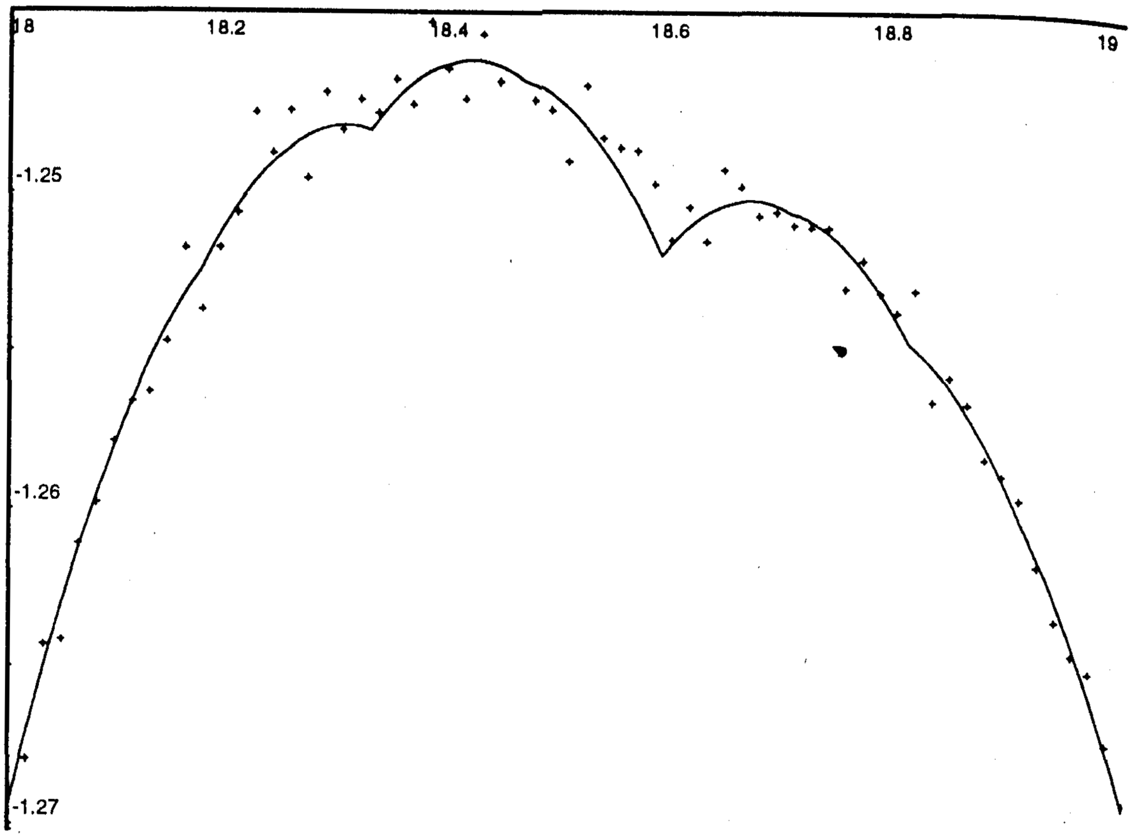

Fig. 8. Display of the limit periodic curve $B(u)$ for $u \in [18, 19]$. Each cross represents one simulation of the cost X_n for $n \in [2^{18}, 2^{19}]$, with a logarithmic scale for n and the usual normalization $(X_n - n \lg n)/n$; X_n was simulated by running top-down mergesort on one random file of n-elements and counting the comparisons performed

Proof. (Sketch) Recall that, in Sect. 1, we showed that the cost of mergesort is the sum of independent random variables where each random variable is the cost of a particular merge; from (1), the variable part of each individual merge cost is found to have a third moment bounded by an absolute constant and a variance of $O(1)$. The proof then directly follows from Lyapounov's generalization of the central limit theorem [5, p. 371]. □

In particular the cost is very close to its average estimate with high probability. For $n = 100$, the probability generating function is of the form

$$6.17683\, 10^{-141} \cdot x^{316} + \ldots + 3.84796\, 10^{-14} \cdot x^{573}.$$

This shows numerically that both of the extreme cases (the best cost of 316 and the worst cost of 573) are highly unlikely. Without getting into further details we mention that the distribution of X_n also admits superexponential tails. That is, the probability that $|X_n - U(n)|$ is large falls off as the distribution of a Gaussian random variable.

The concentration property for the distribution is further illustrated by the simulation data of Fig. 8. Notice that, thanks to a "self averaging" property of Mergesort, we can even verify our theorems by *using samples of size* 1! The (fractal) periodic functions are thus far from being an artifact of our analysis but closely mirror the reality of the algorithm's behavior.

6 Conclusion

Divide-and-conquer recurrences are naturally associated with Dirichlet series that satisfy various sorts of functional relations [1, 2] and that can be proven to have meromorphic continuations in the whole of the complex plane. As we have seen here and as in [11], the Mellin-Perron formula then normally allows us to recover asymptotic properties of the original sequence. Several complications may however occur, and we offer a brief comment.

First, the intervening Dirichlet series are often not explicit, and one has to operate with infinite functional relations. One such example is the Thue-Morse sequence that appears in [12] in connection with a probabilistic estimation algorithm. The Thue-Morse sequence is defined as $\varepsilon_n=(-1)^{\nu(n)}$, where $\nu(n)$ designates the sum of digits of the binary representation of n. Sequences such as these lead to infinite functional equations and integral representations and are typical of the forms which have to be dealt with in more general cases [2].

Another problem is that each sequence has a certain degree of "smoothness" that dictates a certain level of summations. For mergesort, we were able to operate with the Mellin-Perron formula relative to double sums and the integrals we had to evaluate were nicely convergent. In general, this need not be the case. Take for example the cost of Karatsuba multiplication,

$$K(n)=3K\left(\left\lceil\frac{n}{2}\right\rceil\right)+n.$$

From the defining equation, the Dirichlet series of *first* differences has an explicit form,

$$\sum_{n=1}^{\infty}\frac{\Delta K_n}{n^s}=\frac{\zeta(s)}{1-3\cdot 2^{-s}}.$$

In this case the suitable form of the Mellin-Perron formula involves a different kernel with a denominator of the form $1/s$ instead of the $1/(s(s+1))$ that we have encountered so far (and was discussed in Lemma 1), i.e.

$$\frac{K(n)+K(n+1)}{2}=\frac{1}{2i\pi}\int_{3-i\infty}^{3+i\infty}\frac{\zeta(s)n^s}{1-3\cdot 2^{-s}}\frac{ds}{s}.$$

This poses specific convergence problems. Accordingly, the sequence exhibits a discontinuous behavior, for instance $K(2^n+1)/K(2^n)\to 5/2$. In that case, it is the sum $\sum_{n=1}^{N} K(n)$ that appears to be amenable to our treatment: see the closely related example of "triadic binary numbers" in [11].

Acknowledgements. The work of both authors was supported in part by the Basic Research Action of the E.C. under contracts No. 3075 and No. 7141 (Projects ALCOM I and ALCOM II). The work of the second author was also partially supported by NSF grant CCR-81918152 and was performed while at INRIA-Rocquencourt. The authors would also like to thank the anonymous referee who improved the error bound in Theorem 4 from $O(n^\varepsilon)$ to $O(1)$.

References

1. Allouche, J.-P.: Automates finis en théorie des nombres. Expo. Math. **5**, 239–266 (1987)
2. Allouche, J.-P., Cohen, H.: Dirichlet series and curious infinite products. Bull. Lond. Math. Soc. **17**, 531–538 (1985)
3. Allouche, J.-P., Shallit, J.: The ring of k-regular sequences. Theor. Comput. Sci. **98**, 163–197 (1992)
4. Apostol, T.M.: Introduction to analytic number theory. Berlin, Heidelberg, New York: Springer 1976
5. Billingsley, P.: Probability and measure, 2nd edn. New York: John Wiley 1986
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to algorithms. New York: MIT Press 1990
7. Delange, H.: Sur la fonction sommatoire de la fonction somme des chiffres. Enseign. Math. **21** (1), 31–47 (1975)
8. Dumas, P.: Critères de B-régularité. Cah. Sémin. Théorie Nombres (Bordeaux) (1994, to appear)
9. Dumont, J.-M., Thomas, A.: Systèmes de numération et fonctions fractales relatifs aux substitutions. Theor. Comput. Sci. **65**, 153–169 (1989)
10. Flajolet, P., Golin, M.: Exact asymptotics of divide-and-conquer recurrences. In: Lingas, S.C.A., Karlsson, R. (eds.) Automata, languages, and programming (Lect. Notes Comput. Sci., vol. 700, pp. 137–149) Berlin, Heidelberg, New York: Springer 1993
11. Flajolet, P., Grabner, P., Kirschenhofer, P., Prodinger, H., Tichy, R.: Mellin transforms and asymptotics: Digital sums. Theor. Comput. Sci. **123**, 291–314 (1994)
12. Flajolet, P., Martin, G.N.: Probabilistic counting algorithms for data base applications. J. Comput. Syst. Sci. **31** (2), 182–209 (1985)
13. Knuth, D.E.: The art of computer programming, vol. 1: Fundamental algorithms, 2nd edn. Reading, MA: Addison-Wesley 1968
14. Knuth, D.E.: The art of computer programming, vol. 3: Sorting and searching. Reading, MA: Addison-Wesley 1973
15. Sedgewick, R.: Algorithms, 2nd edn. Reading, MA: Addison-Wesley 1988
16. Stolarsky, K.B.: Power and exponential sums of digital sums related to binomial coefficients. SIAM J. Appl. Math. **32** (4) 717–730 (1977)
17. Vardi, I.: Computational recreations in mathematica. Reading, MA: Addison Wesley 1991