LNCS 700 A. Lingas R. Karlsson S. Carlsson Eds.
Automata, Languages, and Programming

Exact Asymptotics of Divide-and-Conquer Recurrences

20th. Intl Colloquium ICALP 93 Lund, Sweden, July 1993
Proceedings Springer-Verlag

Philippe FLAJOLET[1] and Mordecai GOLIN[2]

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

Abstract. The divide–and–conquer principle is a majoı paradigm of algorithms design. Corresponding cost functions satisfy recurrences that directly reflect the decomposition mechanism used in the algorithm.
This work shows that periodicity phenomena, often of a fractal nature, are ubiquitous in the performances of these algorithms. Mellin transforms and Dirichlet series are used to attain precise asymptotic estimates. The method is illustrated by a detailed average case, variance and distribution analysis of the classic top–down recursive mergesort algorithm.
The approach is applicable to a large number of divide–and–conquer recurrences, and a general theorem is obtained when the partitioning–merging toll of a divide–and–conquer algorithm is a sublinear function. As another illustration the method is also used to provide an exact analysis of an efficient maxima-finding algorithm.

Many algorithms are based on a recursive *divide–and–conquer* strategy. Accordingly, their complexity is expressed by recurrences of the usual divide–and–conquer form [10]. Typical examples are heapsort, mergesort, 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 [10, p. 62]: if for example

$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n \tag{1}$$

and $e_n = O(n)$ then $f_n = O(n \log n)$ while if $e_n = O(n^{1-\epsilon})$ for some $\epsilon > 0$ then $e_n = O(n)$. 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. We will show for example that the average number of comparisons performed by mergesort satisfies

$$U(n) = n \lg n + nB(\lg n) + O(n^{1/2}),$$

while the variance is of the form $nC(\lg n) + O(n^{1/2})$: $B(u)$ and $C(u)$ are both periodic functions that are *fractal*–like and which are everywhere continuous but not differentiable at a dense set of points on the line.

Our approach consists in introducing for this range of problems techniques – Mellin transforms, Dirichlet series, and Perron's formula – that are borrowed from classical analytic number theory [4]. These techniques lead to *exact* analyses. For example, we find exact formulas for the functions $B(u)$ and $C(u)$ above. They are of a very wide applicability in this range of problems, a fact that we demonstrate by applying the techniques to the analysis of a maxima finding algorithm in multidimensional space.

The general character of the results attained is attested by Theorem 9. This theorem gives the precise asymptotic form of solutions to divide and conquer recurrences of the form (1), when the partitioning (or dually merging) cost is sublinear.

This paper is only an extended abstract of a full article [15].

1 Mergesort

First, we recall the schema of the **Mergesort** algorithm.

```
Algorithm MergeSort($a[1\,..\,n]$);
• MergeSort($a[1\,..\,\lfloor n/2\rfloor]$);
• MergeSort($a[\lfloor n/2\rfloor+1\,..\,n]$);
• Merge($a[1\,..\,\lfloor n/2\rfloor]$ , $a[\lfloor n/2\rfloor+1\,..\,n]$);
```

Let $T(n)$ denote the worst time cost measured in the number of comparisons that are required for sorting n elements by the MergeSort procedure, and let $U(n)$ be the corresponding average cost. We have

$$T(n)=T(\lfloor\frac{n}{2}\rfloor)+T(\lceil\frac{n}{2}\rceil)+n-1, \qquad U(n)=U(\lfloor\frac{n}{2}\rfloor)+U(\lceil\frac{n}{2}\rceil)+n-\gamma_n \quad (2)$$

for $n\geq 2$, with $T(1)=U(1)=0$, and $\gamma_n=\frac{\lfloor\frac{n}{2}\rfloor}{\lceil\frac{n}{2}\rceil+1}+\frac{\lceil\frac{n}{2}\rceil}{\lfloor\frac{n}{2}\rfloor+1}$. This results from the cost of merging two files of size a and b which is

$$a+b-1 \text{ and } a+b-\frac{a}{b+1}-\frac{b}{a+1},$$

in the worst case and average cases respectively (see [19, p. 165] for a fuller description of recursive mergesort and [18, ex. 5.2.4-2] for a derivation of the average case cost of merging).

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 $\lg n\equiv\log_2 n$. (Recall the usual notation for fractional parts, $\{u\}=u-\lfloor u\rfloor$.) The periodicities are apparent from Fig. 1 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

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

(See [17, 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. □.

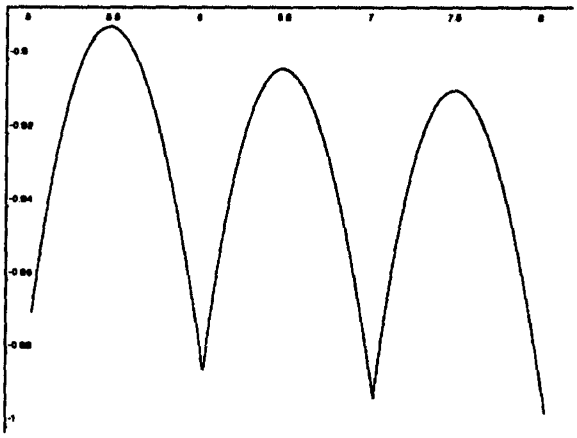

Fig. 1. The fluctuation in the worst case behavior of Mergesort, in the form of the coefficient of the linear term, $\frac{1}{n}[T(n) - n\lg n]$, as a function of $\lg n \equiv \log_2 n$ for $n = 32..256$. From Theorems 1 and 2, the periodic function involved, $A(u)$, fluctuates in $[-1, -0.91392,]$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

Knuth analyzes a bottom up version of Mergesort in the average case (Algorithm L, see [18, 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 $\beta = -\sum_{j\geq 0}\frac{1}{2^j+1} = -1.26449\,97803$.

For general n, no such formula is known. (See however Equation (13) at the end of Section 4 for some related analyses.) 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 approach the analysis of $T(n)$ and $U(n)$ via the computation of some 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 2 (Mellin–Perron). *Assume the Dirichlet series $W(s)$ converges absolutely for $\Re(s) > 2$. Then,*

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

The proof is based on contour integration and the residue theorem; see [4, p. 243] for a closely related result in the context of classic analytic number theory. 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 will determine the Dirichlet series of their second differences. Then we will 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 combinatorial 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.

Consider the general divide-and-conquer recurrence scheme
$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n, \tag{4}$$
for $n \geq 2$, 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 $U(n)$, $e_n = n - \gamma_n$. Take the backward differences $\nabla f_n = f_n - f_{n-1}$ and $\nabla e_n = e_n - e_{n-1}$ and then the double differences (forward of backward) $\Delta\nabla f_n = \nabla f_{n+1} - \nabla f_n$ and $\Delta\nabla e_n = \nabla e_{n+1} - \nabla e_n$. Working through the details we find
$$\begin{cases} \Delta\nabla f_{2m} &= \Delta\nabla f_m + \Delta\nabla e_{2m} \\ \Delta\nabla f_{2m+1} &= \qquad\qquad \Delta\nabla e_{2m+1}, \end{cases} \tag{5}$$
for $m \geq 1$, with $\Delta\nabla f_1 = f_2 - 2f_1 = e_2 = \Delta\nabla e_1$.

Define the Dirichlet generating function corresponding to $w_n = \Delta\nabla f_n$,
$$W(s) = \sum_{n=1}^{\infty} \frac{\Delta\nabla f_n}{n^s}. \tag{6}$$

Then, from (5), multiplying w_n by n^{-s}, summing over n, and solving for $W(s)$, we attain the explicit form
$$W(s) = \frac{1}{1-2^{s}}\left(\Delta\nabla f_1 + \sum_{n=2}^{\infty}\frac{\Delta\nabla e_n}{n^s}\right). \tag{7}$$

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 :

Lemma 3. *Consider the recurrence*
$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + e_n,$$
for $n \geq 2$, with f_1 given and $e_n = O(n)$. The solution satisfies
$$f_n = nf_1 + \frac{n}{2i\pi}\int_{3-i\infty}^{3+i\infty} \frac{\Xi(s)n^s}{1-2^{-s}}\frac{ds}{s(s+1)},$$
where $\Xi(s) = \sum_{n=1}^{\infty} \frac{\Delta\nabla e_n}{n^s}$.

$\Delta\nabla f_1 +$ $= \sum_{m=2}^{\infty} e_m\left((m+1)^{-s} - 2m^{-s} + (m-1)^{-s}\right)$

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

3 Worst Case of Mergesort

As an application of Lemma 3 we quickly sketch how it can be used to derive an alternate expression involving a Fourier series for the value $T(n)$, the *worst case* number of comparisons performed by mergesort.

Theorem 4. *The worst case cost $T(n)$ satisfies*

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

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, $A(u) = \sum_{k\in\mathbb{Z}} a_k e^{2ik\pi u}$, where, for $k \in \mathbb{Z} \setminus \{0\}$,

$$a_k = \frac{1}{\log 2}\frac{1}{\chi_k(\chi_k+1)} \qquad \textit{with } \chi_k = \frac{2ik\pi}{\log 2}.$$

The extreme values of $A(u)$ are

$$-\frac{1+\log\log 2}{\log 2} = -0.91392, \textit{and } \; -1.$$

Proof. We apply Lemma 3 with $f_n = T(n)$. For this case we have $e_n = n-1$ and $f_1 = 0$ so $\Delta\nabla e_1 = e_2 = 1$ and $\Delta\nabla e_n = 0$ for all $n \geq 2$. Thus $\Xi(s) = 1$ and

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

We can evaluate this integral using residue computations. 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

$$\begin{aligned} \Gamma_1 &= \{3+iy : |y| \leq R\}, & \Gamma_2 &= \{x+iR : \alpha \leq x \leq 3\}, \\ \Gamma_3 &= \{\alpha+iy : |y| \leq R\}, & \Gamma_4 &= \{x-iR : \alpha \leq x \leq 3\}. \end{aligned} \tag{9}$$

(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.) Set $I(s) = \frac{n^s}{1-2^{-s}}\frac{1}{s(s+1)}$ to be the kernel of the integral in (8). Letting $R \uparrow \infty$ we find that $\frac{1}{2i\pi}\int_{\Gamma_1} I(s)\,ds$ becomes the integral in (8), $|\int_{\Gamma_2} I(s)ds|$ and $|\int_{\Gamma_4} I(s)ds|$ are both $O\left(1/R^2\right)$ and

$$\left|\int_{\Gamma_3} I(s)ds\right| \to \left|\int_{\alpha+i\infty}^{\alpha-i\infty} I(s)ds\right| \leq 4n^\alpha.$$

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 $s = 2ki\pi/\log 2$, $k \in \mathbb{Z}\setminus\{0\}$ with residues $a_k e^{2ik\pi \lg n}$.

Thus, as promised, we have shown that $T(n) = n\lg n + nA(\lg n) + 1$. □.

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). However, the calculations performed above are needed in the analysis of the average case behavior in the next section.

4 Average Case of Mergesort

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

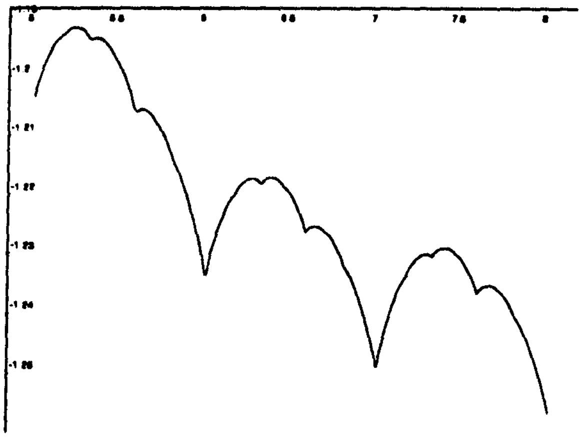

Fig. 2. The fluctuation in the average case behavior of Mergesort, graphing the coefficient of the linear term, $\frac{1}{n}[U(n) - n\lg n]$, using a logarithmic scale for $n = 32..256$. From Theorem 3, the periodic function involved, $B(u)$, fluctuates in $[-1.26449, -1.24075]$ with mean value $b_0 = -1.24815$.

Theorem 5. *(i). Let $\epsilon > 0$. The average case cost $U(n)$ of Mergesort satisfies*

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

where $B(u)$ is periodic with period 1 and everywhere continuous but non-differentiable at every point $u = \{\lg n\}$. Furthermore, B has an explicit Fourier expansion.

(ii). The mean value $b_0 = -1.24815\,20420\,99653\,88489\ldots$ of $B(u)$ is

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

(iii). $B(u) = \sum_{k\in\mathbb{Z}} b_k e^{2ik\pi u}$ where b_0 is as above and the other Fourier coefficients of $B(u)$ are, for $k \in \mathbb{Z}\setminus\{0\}$,

$$b_k = \frac{1}{\log 2}\frac{1+\Psi(\chi_k)}{\chi_k(\chi_k+1)} \qquad \textit{where } \chi_k = \frac{2ik\pi}{\log 2},$$

and

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

This Fourier series is uniformly convergent to $B(u)$.

(iv). The extreme values of $B(u)$ are

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

Proof. The proof follows the paradigm laid down by Theorem 5. We first use Lemma 3 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 $\Delta\nabla e_1 = e_2 = 1$. We are also given that for all $m > 0$

$$\begin{cases} e_{2m} &= 2m - 2 + \frac{2}{m+1} \\ e_{2m+1} &= 2m - 1 + \frac{2}{m+2}, \end{cases} \tag{10}$$

and thus

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

Summing over all m we may write $\Xi(s) = \Delta\nabla e_1 + \sum_{n=2}^{\infty} \frac{\Delta\nabla e_n}{n^s} = 1 + \Psi(s)$ 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 $\Re(s) = \alpha \geq -1 + \epsilon$. Lemma 3 therefore tells us that

$$\frac{f_n}{n} = \frac{1}{2i\pi} \int_{3-i\infty}^{3+i\infty} \frac{n^s}{1-2^{-s}} \frac{ds}{s(s+1)} + \frac{1}{2i\pi} \int_{3-i\infty}^{3+i\infty} \frac{n^s \Psi(s)}{1-2^{-s}} \frac{ds}{s(s+1)}. \tag{11}$$

The first integral on the right-hand side was already evaluated during the proof of Theorem 4 and shown to be equal to $\lg n + A(\lg n) + 1$ where $A(u) = \sum_k a_k 2^{ik\pi u}$.

The second integral can be evaluated using similar techniques (details omitted).

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

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

both functions having mean value 0. By exchanging summations, we find

$$B^*(u) - A^*(u) = \sum_{m=1}^{\infty} \psi_m A^*(u - \lg m), \tag{12}$$

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

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

This unusual decomposition (12) explains the behavior of $U(n)$ in Fig. 2. 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. 2. 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 [21]. We have $\Psi'(0) = \sum_{m=1}^{\infty} \theta(1/m)$, for some function $\theta(y)$ analytic at the origin. Such sums can be transformed into fast converging sums involving the Riemann zeta function, $\zeta(s) = \sum_{n\geq 1} n^{-s}$. In this way, we evaluate $\Psi'(0)$ to 50 digits in a matter of one minute of computation time.

Extreme values. Regarding the computation of *extreme values* of $B(u)$ accurately, the approach via the Fourier series does not seem to be practicable, since the Fourier coefficients only decrease as $O(k^{-2})$. Consider instead the sequence $U(a2^k)$ for some fixed integer a. By unwinding the recurrence, 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 $n = a2^k$, and taking care of the error terms yields for these particular values of n,

$$U(n) = n \lg n + \beta(a)n + o(n) \quad \text{where} \quad \beta(a) = \frac{U(a)}{a} - \lg a - \sum_{j=0}^{\infty} \frac{1}{a2^j + 1}. \qquad (13)$$

This formula is a *real* formula that generalizes the one given by Knuth for the average case, when $n = 2^k$. Comparing with Theorem 3, we find that

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

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

From these estimates, Mergesort has been found to have an average case complexity about

$$n \lg n - (1.25 \pm 0.01)n + o(n).$$

This is not far from the information theoretic lower bound,

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

5 Variance of Mergesort

The cost of Mergesort is the sum of the costs of the individual merges, which are independent random variables with a known distribution. Merging two files of size m and n costs $m + n - S$, where the random variable S has distribution [18, p. 620]

$$\Pr\{S \geq s\} = \frac{\binom{m+n-s}{m} + \binom{m+n-s}{n}}{\binom{m+n}{n}}. \qquad (14)$$

Then, the variance $V(n)$ of Mergesort applied to random data of size n is a solution to the another divide-and-conquer recurrence. Applying Lemma 3 we find:

Theorem 6. *The variance of the MergeSort algorithm applied to data of size n satisfies*

$$V(n) = n \cdot C(\log_2 n) + o(n),$$

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 $c_0 \approx 0.34549\,95688$.

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 $[0.30, 0.36]$.

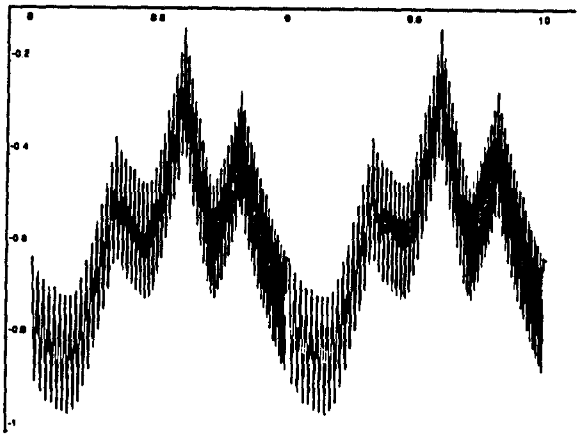

Fig. 3. The clearly fractal fluctuation in the best case behavior of Mergesort, graphing the coefficient of the linear term $\frac{1}{n}[Y(n) - \frac{1}{2} n \lg n]$ using a logarithmic scale for $n = 256 \,..\, 1024$.

6 Best Case of Mergesort

The best case of a merge occurs each time all elements in the larger file dominate the largest element of the smaller file. Thus, the quantity $Y(n)$ representing the smallest number of comparisons—the *best case*—of mergesort satisfies the divide and conquer recurrence:

$$Y(n) = Y(\lfloor \frac{n}{2} \rfloor) + Y(\lceil \frac{n}{2} \rceil) + \lfloor \frac{n}{2} \rfloor. \tag{15}$$

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

$$Y(n) = \sum_{m \le n} \nu(m). \tag{16}$$

Equation (16) has been already noticed by several authors (see, e.g., [3]). The function $Y(n)$ has been studied by Delange [11] using elementary real analysis. It can also be subjected to the methods of this paper (see [16] for a discussion of exact summatory formulæ), and one gets:

Theorem 7. *The best case cost $Y(n)$ satisfies*

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

where $D(u)$ has Fourier coefficients $d_0 = \lg\sqrt{\pi} - \frac{1}{2\log 2} - \frac{1}{4}$,

$$d_k = -\frac{1}{\log 2}\frac{\zeta(\chi_k)}{\chi_k(\chi_k+1)}, \quad k \neq 0, \; \chi_k = \frac{2ik\pi}{\log 2}.$$

Delange already proved that the periodic function $D(n)$ is continuous but nowhere differentiable.

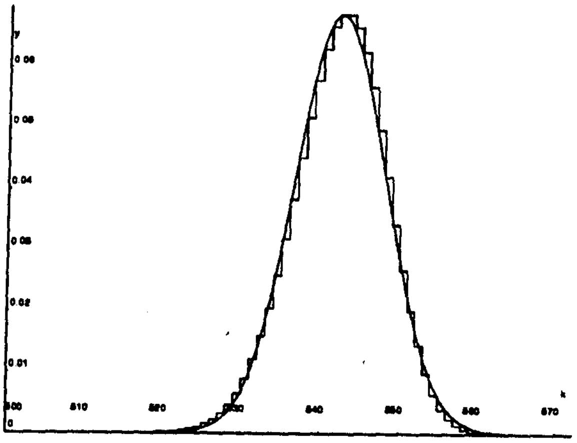

Fig. 4. The histogram of the exact probability distribution of the comparison cost of Mergesort for $n = 100$ and its fitting Gausian curve.

7 Distribution of Mergesort

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 of the single merge intervening in the sorting of n elements is found from (14). The probability generating function of the cost of merge sort then satisfies the divide–and–conquer product recurrence,

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

Unwinding the recurrence yields

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

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..573] with mean value 541.84. The standard deviation is 5.78, and Figure 4 shows the histogram of the distribution computed from these formulæ. The numerical

$$\xi_n(z) = z^n + (1-z) \sum_{k \geq 1} \frac{\binom{n-k}{\lfloor n/2 \rfloor} + \binom{n-k}{\lceil n/2 \rceil}}{\binom{n}{\lfloor n/2 \rfloor}} z^{n-k}$$

data strongly suggest convergence to a Gaussian law with matching mean and variance that is also plotted on the same diagram.

Actually, using Lyapounov's extensions of the central limit theorem [8, p. 371] to sums of independent—but not necessarily identically distributed—random variables, we find:

Theorem 8. *The cost X_n of Mergesort applied to random data of size n converges in distribution to a normal variable,*

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

8 Maxima-Finding

The tools that we have developed in the preceding sections are very general and can be used to analyze a large number of divide-and-conquer type algorithms. The following theorem precisely quantifies the behaviour of most linearly growing recurrences occurring in practice.

Theorem 9. *Assume that for some $\epsilon > 0$ the series $(\sum_n \Delta\nabla e_n \cdot n^{\epsilon})$ converges absolutely. Then*

$$f_n = nQ(\lg n) + O(n^{1-\epsilon}),$$

where $Q(u)$ is periodic and fractal, with mean value

$$q_0 = \frac{1}{\log 2}\sum_{n=2}^{\infty} e_n \log(\frac{n^2}{n^2-1}).$$

As an application of this theorem, we briefly sketch how to analyze the expected running time of a maxima-finding algorithm. The interesting feature of this analysis is that the running time will grow as $nQ(\lg n)$ where $Q(u)$ is a continuous, fractal like function which is non differentiable. Thus here, unlike in the running time of mergesort, the periodic term appears in the highest order asymptotics.

A d-dimensional point $P = (p_{(1)}, \dots p_{(d)})$ *dominates* a point $Q = (q_{(1)}, \dots q_{(d)})$ if $P \neq Q$ and $p_{(j)} \geq q_{(j)}$. A maximal element of a finite set $\{P_1, \dots, P_n\}$ is a point in the set which is not dominated by any other point in the set. Maximal elements are of interest for a variety of reasons and much work has therefore been done on devising algorithms to identify them, e.g. [6] [7]. One of these algorithms is the divide–and–conquer one discussed by [12]: given a set of n points, split the set into two subsets of size $\lfloor n/2 \rfloor$, $\lceil n/2 \rceil$, recursively find the maxima in each of the subsets and then determine the maxima of the entire set by pairwise comparisons of all the maxima in the first subset to all of the maxima in the second.

It is known [9] that if n points are drawn independently identically distributed (IID) from the uniform distribution over a hypercube or, in fact, from any component-independent distribution then the expected number of the points that will be maximal is

$$\mu_n^{(d)} = \sum_{k_{d-1}=1}^{n} \frac{1}{k_{d-1}} \sum_{k_{d-2}=1}^{k_{d-1}} \frac{1}{k_{d-1}} \cdots \sum_{k_2=1}^{k_3} \frac{1}{k_2} \sum_{k_1=1}^{k_2} \frac{1}{k_1}.$$

$X_n = X_{\lfloor n/2 \rfloor} + X_{\lceil n/2 \rceil} + A_n B_n \Rightarrow EX_n = EX_{\lfloor n/2 \rfloor} + EX_{\lceil n/2 \rceil} + \underset{EA_n \, EB_n}{\underset{\shortparallel}{E(A_n B_n)}}$

For example $\mu_n^{(2)} = H_n = \sum_{k \le n} 1/k$ is the harmonic number. The average running time of the divide-and-conquer maxima finding algorithm when run on inputs chosen IID from a d-dimensional hypercube (or component independent distribution) therefore satisfies $f_1 = 1$ with

$$f_n = f_{\lfloor n/2 \rfloor} + f_{\lceil n/2 \rceil} + 2\mu^{(d)}_{\lfloor n/2 \rfloor} \cdot \mu^{(d)}_{\lceil n/2 \rceil} \tag{17}$$

for $n \ge 2$.

It is not difficult to see that $\mu_n^{(d)} = O\big(\log^{d-1} n\big)$ so we find automatically (as is done in [12]) that $f_n = O(n)$. Observe that this seemingly naïve algorithm has linear expected case, and thus beats a simple sweepline algorithm already in dimension $d = 2$, the latter requiring sorting. Using the techniques introduced earlier in this paper, we can go much further and derive the *exact* asymptotics of f_n.

Theorem 10. *Let $\epsilon > 0$. The expected running time of the maxima finding algorithm when run on inputs chosen IID from a d-dimensional hypercube satisfies*

$$f_n = nQ^{(d)}(\lg n) + O\left(n^{\epsilon}\right)$$

where $Q^{(d)}(u)$ is a continuous, periodic, non-differentiable function with mean value

$$q_0^{(d)} = 2\sum_{m=1}^{\infty} (\mu_m^{(d)})^2 \log(1-(2m)^{-2})^{-1} + 2\sum_{m=1}^{\infty} \mu_m^{(d)}\mu_{m+1}^{(d)} \log(1-(2m+1)^{-2})^{-1},$$

and $\mu_n^{(d)} = [u^{d-1}]\exp(\frac{u}{1}H_n^{(1)} + \frac{u^2}{2}H_n^{(2)} + \cdots)$.

With computer algebra, the mean values can be calculated to high accuracy using relations between the Dirichlet series of generalized harmonic numbers and derivatives of the Riemann Zeta function [5], as well as the techniques discussed previously. For example, we have

$$q_0^{(2)} = 6.32527\ldots, \quad q_0^{(3)} = 21.64397\ldots, \quad q_0^{(4)} = 76.77212\ldots.$$

9 Conclusion

Divide–and–conquer recurrences are naturally associated with Dirichlet series that satisfy various sorts of functional relations (see also the case of 'automatic' sequences in [1, 2, 13]) so that they can be proven to have continuations in the whole of the complex plane. As we have seen here and as in [16], the Mellin–Perron formula then allows us to recover asymptotic properties of the original sequence with great accuracy, revealing periodicities and fractal behaviour for these recurrences.

ACKNOWLEDGEMENTS: The work of both authors was supported in part by the Basic Research Action of the E.C. under contract No. 7141 (Project ALCOM II). the second author was also supported by NSF CCR-8918152.

Authors' electronic mail addresses: `Philippe.Flajolet@inria.fr` and `golin@cs.ust.hk`.

References

1. Allouche, J.-P. Automates finis en théorie des nombres. *Expositiones Mathematicae 5* (1987), 239–266.
2. Allouche, J.-P., and Cohen, H. Dirichlet series and curious infinite products. *Bulletin of the London Mathematical Society 17* (1985), 531–538.
3. Allouche, J.-P., and Shallit, J. The ring of k-regular sequences. *Theoretical Computer Science 98* (1992), 163–197.
4. Apostol, T. M. *Introduction to Analytic Number Theory.* Springer-Verlag, 1976.
5. Apostol, T. M., and Vu, T. H. Dirichlet series related to the Riemann zeta function. *Journal of Number Theory 19* (1984), 85–102.
6. Bentley, J. L., Clarkson, K. L., and Levine, D. B. Fast linear expected-time algorithms for computing maxima and convex hulls. In *First Symposium on Discrete Algorithms (SODA)* (1990).
7. Bentley, J. L., Kung, H., Schkolnick, M., and Thompson, C. On the average number of maxima in a set of vectors and applications. *Journal of the Association for Computing Machinery 25*, 4 (October 1978), 536–543.
8. Billingsley, P. *Probability and Measure*, 2nd ed. John Wiley & Sons, 1986.
9. Buchta, C. On the average number of maxima in a set of vectors. *Information Processing Letters 33* (Nov. 1989), 63–65.
10. Cormen, T. H., Leiserson, C. E., and Rivest, R. L. *Introduction to Algorithms.* MIT Press, New York, 1990.
11. Delange, H. Sur la fonction sommatoire de la fonction somme des chiffres. *L'enseignement Mathématique XXI*, 1 (1975), 31–47.
12. Devroye, L. Moment inequalities for random variables in computational geometry. *Computing 30* (1983), 111–119.
13. Dumas, P. *Récurrences Mahlériennes, suites automatiques, et études asymptotiques.* Doctorat de mathématiques, Université de Bordeaux I, 1992. In preparation.
14. Dumont, J.-M., and Thomas, A. Systèmes de numération et fonctions fractales relatifs aux substitutions. *Theoretical Computer Science 65* (1989), 153–169.
15. Flajolet, P., and Golin, M. Mellin transforms and asymptotics: The mergesort recurrence. Preprint submitted to *Acta Informatica*, Jan. 1993.
16. Flajolet, P., Grabner, P., Kirschenhofer, P., Prodinger, H., and Tichy, R. Mellin transforms and asymptotics: Digital sums. Research Report 1498, Institut National de Recherche en Informatique et en Automatique, Sept. 1991. 23 pages. To appear in *Theoretical Computer Science*, December 1993.
17. Knuth, D. E. *The Art of Computer Programming*, vol. 1: Fundamental Algorithms. Addison-Wesley, 1968. Second edition, 1973.
18. Knuth, D. E. *The Art of Computer Programming*, vol. 3: Sorting and Searching. Addison-Wesley, 1973.
19. Sedgewick, R. *Algorithms*, second ed. Addison–Wesley, Reading, Mass., 1988.
20. Stolarsky, K. B. Power and exponential sums of digital sums related to binomial coefficients. *SIAM Journal on Applied Mathematics 32*, 4 (1977), 717–730.
21. Vardi, I. *Computational Recreations in Mathematica.* Addison Wesley, 1991.