BOTTOM-UP MERGESORT[†]
A Detailed Analysis

W. PANNY[1] and H. PRODINGER[2]

[1] *Department of Computer Science, University of Economics, Augasse 2–6, A-1090 Vienna, Austria.*
E-mail: `wolfgang.panny@wu-wien.ac.at`

[2] *Department of Algebra and Discrete Mathematics, Technical University, Wiedner Hauptstrasse 8–10, A-1040 Vienna, Austria.*
E-mail: `proding@email.tuwien.ac.at`

Abstract: Though the behaviors of mergesort algorithms are basically known, the periodicity phenomena encountered in their analyses are not easy to deal with. In this paper closed-form expressions for the necessary number of comparisons are derived for the bottom-up algorithm, which adequately describe its periodic behavior. This allows, among other things, to compare the top-down and the bottom-up mergesort algorithms.

Keywords: Sorting, mergesort; analysis of algorithms.

1. Introduction

Mergesort is one of the most prominent internal sorting methods. Especially for *lists* it is one of the best available alternatives (cf. [Knuth], [Gonnet]). Among the particular advantages of mergesort — also compared to *Quicksort* and its variants — it maintains its $O(N \log N)$ time complexity also in the worst case (in fact the worst case time is only $O(N)$ worse than the average time) and it is stable.

There are two basic variants of mergesort, namely *top-down* mergesort and *bottom-up* mergesort. Yet another variant — termed *queue* mergesort — recently has been proposed in [Golin, Sedgewick]. For both of the basic variants the behavior of the quantities determining the running time of the pertaining algorithms is basically known (cf. e.g. [Gonnet, p. 173], [Knuth, ex. 5.2.4-13]). Of these quantities the number $C(N)$ of comparisons necessary to sort N elements is perhaps the most important since it has the strongest impact on running time.

One interesting feature about mergesort is the occurrence of periodicity phenomena — which of course are not uncommon in the analysis of divide-and-conquer algorithms. Let us introduce the notation $C_{b,td}(N)$, $C_{w,td}(N)$ and $C_{a,td}(N)$ for the number of comparisons of the top-down algorithm in the best, worst and average case, respectively. In a recent paper *Flajolet* and *Golin* have shown how to deal with

[†] Accepted for publication in *Algorithmica* (April 5, 1994)

such periodicity phenomena encountered in the analysis of $C_{w,td}(N)$ and $C_{a,td}(N)$. One has for example

$$C_{w,td}(N) = N \lg N + NA(\lg N) + 1, \tag{1}$$

where the periodic function A is defined by $A(x) = 1 - \{x\} - 2^{1-\{x\}}$. It should be evident that this last expression describes the behavior of $C_{w,td}(N)$ better than the conventional expression $\lceil \lg N \rceil N - 2^{\lceil \lg N \rceil} + 1$ since it explicitely reflects the periodic behavior of the linear term. On the other hand, it is well known (cf. [Gonnet, p.173]) that $C_{b,td}(N) = \sum_{n<N} \nu(n)$, where $\nu(n)$ is the number of 1's in the binary representation of n. The last relation reveals a surprising connection with the cumulated sum of digits function known from number theory, whose linear term again shows a periodical fluctuation.

The corresponding quantities $C_{b,bu}(N)$, $C_{w,bu}(N)$ and $C_{a,bu}(N)$ of the bottom-up variant show a similar periodic behavior. In this paper we are analyzing this periodicity in order to derive explicit closed-form expressions similar to (1) for the above quantities. So, in a way, this paper may be seen as a complement to [Flajolet, Golin] for the bottom-up algorithm. Some surprising connections to number theoretic problems will again be encountered. Our results also seem to be of some relevance for more practical considerations since they allow to precisely compare the two mergesort variants.

2. Bottom-Up Mergesort

The basic scheme of bottom-up mergesort may be characterized in the following way:

```
Algorithm MergeSort(a[1..N])
  begin
  length := 1;
  while length < N do
        begin
        p := 0;  q := length;
        while q + length ≤ N do
              begin
              Merge(a[p + 1..p + length],  a[q + 1..q + length]);
              p := q + length;  q := p + length
              end;
        if q < N then Merge(a[p + 1..p + length],  a[q + 1..N]);
        length := 2 * length
        end
  end;
```

The following recursive formulation shows that bottom-up mergesort — like the top-down variant — follows the divide-and-conquer paradigm. Only the splitting strategy is different. It is just this "power of two" splitting strategy — as opposed to the "half and half" strategy of the top-down variant — which makes it possible to devise the above nonrecursive algorithm.

```
Algorithm MergeSort(a[1 .. N])
  begin
  if N ≥ 2
  then begin
       MergeSort(a[1 .. 2^(⌈lg N⌉−1)]);
       MergeSort(a[2^(⌈lg N⌉−1)+1 .. N]);
       Merge(a[1 .. 2^(⌈lg N⌉−1)], a[2^(⌈lg N⌉−1)+1 .. N])
       end
  end;
```

The number of elements to be sorted will be denoted by N with binary representation $N = (b_k b_{k-1} \dots b_1 b_0)_2$. At the bottom level we have $\lfloor N/2 \rfloor$ merge processes of type $(1,1)$. At the next level we have $\lfloor N/4 \rfloor$ merge processes of type $(2,2)$. If $N \bmod 4 > 2$ there is also an "incomplete" merge process of type $(2, N \bmod 2)$. In general, at level j $(j \geq 0)$ there are $\lfloor N/2^{j+1} \rfloor$ complete merge processes of type $(2^j, 2^j)$. If $N \bmod 2^{j+1} > 2^j$ there is an additional incomplete merge process of type $(2^j, N \bmod 2^j)$. Whether such an incomplete merge process occurs at level j is easily seen from the binary representation of N: it takes place if and only if $b_j (b_{j-1} b_{j-2} \dots b_1 b_0)_2 > 0$.

Altogether there are $N-1$ merge processes, viz. $\lfloor N/2 \rfloor + \lfloor N/4 \rfloor + \dots + \lfloor N/2^k \rfloor$ complete merge processes and $b_0 + b_1 + \dots + b_k - 1$ incomplete merge processes. The number of levels necessary to sort N elements is $\lceil \lg N \rceil$.

The cost measures $C_{b,bu}(N)$, $C_{w,bu}(N)$ and $C_{a,bu}(N)$ are, of course, ultimately based on the quantities $c_{n,m}$, denoting the number of comparisons in the single (n,m) merge processes. It is well known (cf. e.g. [Knuth, ex. 5.2.4-2]) that

$$\min\{n, m\} \leq c_{n,m} \leq n + m - 1 \tag{2}$$

and that

$$\mathbf{P}\{c_{n,m} = j\} = \frac{\binom{j-1}{n-1} + \binom{j-1}{m-1}}{\binom{n+m}{n}}, \tag{3}$$

under the usual assumptions. The factorial moments are (cf. [Panny])

$$\mathbf{E}\{c_{n,m}^{r}\} = nm \left(\frac{1}{n+r} + \frac{1}{m+r} \right) [n+m+1]^{r-1}, \tag{4}$$

where $[x]^k = x(x+1)\dots(x+k-1)$. The variables $c_{n,m}$ related to the single merge processes are independent of each other.

2.1 The best Case

The overall best case $C_{b,bu}(N)$ is realized exactly if every single merge process is best. Hence, according to (2), one only has to sum the smaller string lengths over all merge processes:

$$C_{b,bu}(N) = \sum_{0<j\leq k} 2^{j-1} \left\lfloor \frac{N}{2^j} \right\rfloor + \sum_{0<j\leq k} b_j (b_{j-1} \dots b_1 b_0)_2 = \sum_{n<N} \nu(n), \tag{5}$$

where $\nu(n)$ denotes the number of 1's in the binary representation of n. The right hand side equality is easily seen by the following inductive argument: Let $C_{b,bu}(N+1) = C_{b,bu}(N) + \Delta(N)$. The bottom level contributes to $\Delta(N)$ with one additional comparison if N is odd. At the next level we have one additional comparison if $N \bmod 4 \geq 2$. In general, level j contributes to $\Delta(N)$ if and only if $b_j = 1$. Hence $\Delta(N) = \nu(N)$. The periodicity and fractal nature of $C_{b,bu}(N)$ is illustrated by Fig.1.

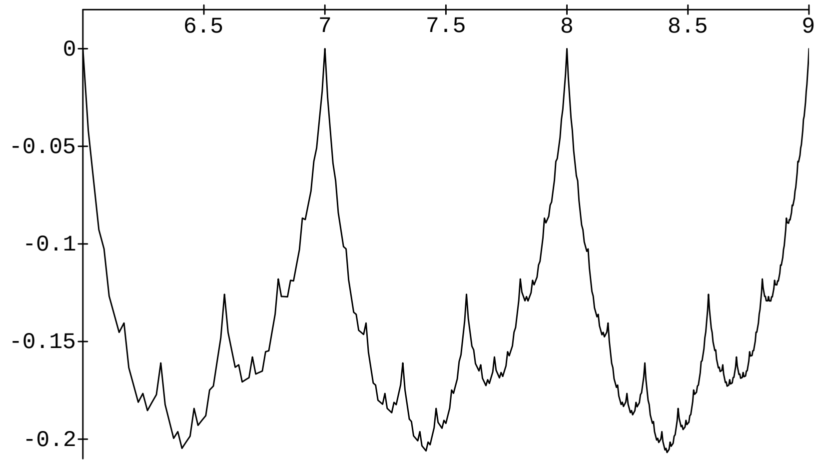

Fig. 1: The fluctuation of the low order terms for the best case, expressed by $(C_{b,bu}(N) - (N \lg N)/2)/N$. Values are plotted against $\lg N$ for $N = 2^6..2^9$.

The behavior of $\sum_{n<N} \nu(n)$ has been extensively studied by [Delange]. The so-called *Trollope-Delange formula* tells us that

$$\sum_{n<N} \nu(n) = \frac{1}{2} N \lg N + N \cdot F_0(\lg N).$$

By appropriately rewriting (5) one sees that $F_0(x) = F_0(\{x\})$ and for $x \in [0,1)$ this function can be defined by

$$F_0(x) = \frac{1}{2^x} \sum_{j \geq 0} \left(\lfloor 2^{x+j} \rfloor - 2 \lfloor 2^{x+j-1} \rfloor - \frac{1}{2} \right) \frac{\{2^{x+j}\}}{2^j} - \frac{x}{2}.$$

Using the binary representation of $2^x = (\beta_0.\beta_1\beta_2\ldots)_2$ this can be written in a more palatable way as

$$F_0(x) = \frac{1}{2^x} \sum_{j \geq 0} \left(\beta_j - \frac{1}{2} \right) (0.\underbrace{0\ldots0}_{j\ 0\text{'s}} \beta_{j+1}\beta_{j+2}\ldots)_2 - \frac{x}{2}.$$

$F_0(x)$ is a continuous, periodic, nowhere differentiable function of fractal nature with an explicit Fourier expansion involving Riemann's zeta function (cf. [Delange], [Flajolet et al.]). The period of F_0 is 1 and its Fourier series

$$F_0(x) = \sum_{k=0,\pm1,\ldots} f_k e^{2\pi i k x}$$

has the coefficients

$$f_0 = \frac{\lg \pi}{2} - \frac{1}{2\log 2} - \frac{1}{4} = -0.145599455\ldots,$$
$$f_k = -\frac{1}{\log 2}\frac{\zeta(\chi_k)}{\chi_k(\chi_k+1)} \qquad \text{for } \ k \neq 0,$$

where

$$\chi_k = \frac{2\pi i k}{\log 2}.$$

The coefficient f_0 tells us that $F_0(x)$ on the average has a value of $-0.145599455\ldots$. The maximum of 0 is attained at the integer points. The minimum of $(\lg 3)/2-1 = -0.207518749\ldots$ is realized in the points congruent to $\lg(4/3)$ (cf.[Foster]). This means that, regarding the fluctuation, the best case is "worst" if N is a power of 2 and is "best" for N next to a point $2^m/3$.

2.2 The worst Case

The overall worst case $C_{w,bu}(N)$ is realized exactly if every single merge process is worst. Applying (2) one accordingly has to sum both string lengths reduced by 1 over all merge processes. Proceeding that way one obtains

$$C_{w,bu}(N) = \sum_{0<j\leq k} 2^j \left\lfloor \frac{N}{2^j} \right\rfloor + \sum_{0<j\leq k} b_j (b_{j-1}b_{j-2}\ldots b_0)_2 - 2^{v_2(N)} + 1, \tag{6}$$

where $v_2(N)$ denotes the number of trailing zeros in the binary representation of N. By properly rewriting (6) one gets the following representation for $C_{w,bu}(N)$

$$N\lg N + N\Big[\sum_{0<j\leq k} (b_j-1)(b_{j-1}b_{j-2}\ldots b_0)_2/N - \{\lg\, N\} - 2^{v_2(N)}/N\Big] + 1\,,$$

which shows that the main term is $N\lg N$ and that the bracketed expression remains constant, when N is replaced by $2N$. Fig. 2 illustrates the periodical fluctuation of the linear term.

The last representation suggests the introduction of the function $F_1(x)$, defined by

$$F_1(x) = F_1(\{x\}) = \frac{1}{2^{\{x\}}}\sum_{j\geq 0}\Big(\big\lfloor 2^{\{x\}+j}\big\rfloor - 2\big\lfloor 2^{\{x\}+j-1}\big\rfloor - 1\Big)\,\frac{\{2^{\{x\}+j}\}}{2^j} - \{x\}.$$

Using again the binary representation of $2^{\{x\}} = (\beta_0.\beta_1\beta_2\ldots)_2$ this becomes

$$F_1(x) = \frac{1}{2^{\{x\}}}\sum_{j\geq 0}\Big(\beta_j - 1\Big)(0.\underbrace{0\ldots0}_{j\ 0\text{'s}}\beta_{j+1}\beta_{j+2}\ldots)_2 - \{x\}.$$

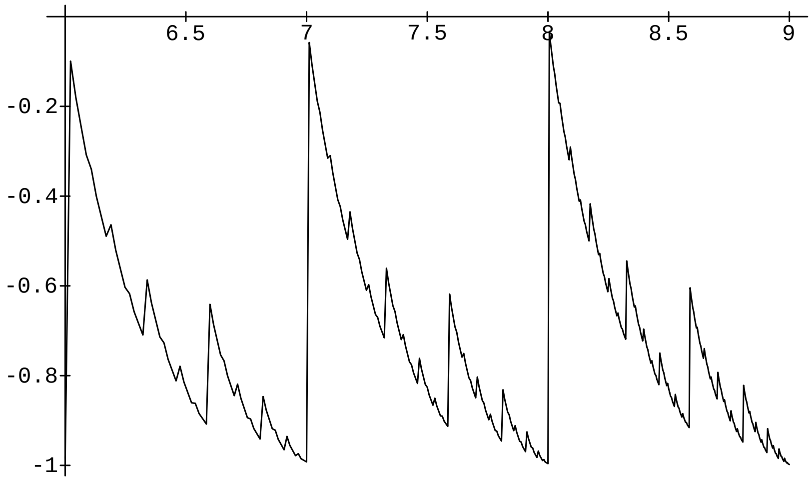

Fig. 2: The fluctuation of the low order terms for the worst case, expressed by $(C_{w,bu}(N) - N \lg N)/N$. Values are plotted against $\lg N$ for $N = 2^6 .. 2^9$.

For the remaining term $2^{v_2(N)}/N$ we may introduce the function $F_2(x)$, defined by

$$F_2(x) = F_2(\{x\}) = \begin{cases} 2^{-m-\{x\}}, & \text{if } 2^{\{x\}} = (1.\beta_1\beta_2 \dots \beta_m)_2 \text{ is dyadic;} \\ 0, & \text{elsewhere.} \end{cases}$$

By means of the functions $F_1(x)$ and $F_2(x)$ the linear term of $C_{w,bu}(N)$ can be expressed by

$$N(F_1(\lg N) - F_2(\lg N)).$$

In the course of his study of binomial queues *Chéno* has encountered the function

$$G(x) = G(\{x\}) = 1 - \{x\} - \frac{1}{2^{\{x\}}} \sum_{j \geq 0} 2^{-j} \{2^{\{x\}+j}\},$$

which is closely related to the functions $F_0(x)$ and $F_1(x)$. Obviously, $F_1(x)$ may be expressed by

$$F_1(x) = F_0(x) + \frac{1}{2} G(x) - \frac{1}{2}.$$

$G(x)$ is periodic with period 1 and has an explicit Fourier series with coefficients (cf. [Chéno])

$$g_0 = \frac{1}{2} + \frac{\gamma - 1}{\log 2} = -0.1099488636 \dots,$$
$$g_k = \frac{1}{\log 2} \frac{\zeta(1 + \chi_k)}{1 + \chi_k} \qquad \text{for} \quad k \neq 0,$$

where again

$$\chi_k = \frac{2\pi i k}{\log 2}.$$

Since G is not continuous we can not assure the convergence of its Fourier series, hence — apart from g_0 — the coefficients do not make too much sense. But they enable us to determine the coefficients of the Fourier series

$$F_1(x) = \sum_{k=0,\pm1,\dots} c_k e^{2\pi i k x},$$

viz.

$$\begin{aligned} c_0 &= \frac{\lg \pi}{2} - \frac{2-\gamma}{2\log 2} - \frac{1}{2} = -0.7005738875\dots, \\ c_k &= \frac{1}{(\chi_k+1)\log 2}\Big(\frac{\zeta(\chi_k+1)}{2} - \frac{\zeta(\chi_k)}{\chi_k}\Big) \qquad \text{for} \quad k \neq 0. \end{aligned}$$

By examining more closely the F_1-function one sees that it is continuous except at points α, where $2^{\{\alpha\}} = (1.\beta_1\beta_2\dots\beta_m)_2$ is dyadic. In these points F_1 is only continuous from the right with jumps of $2^{-m-\{\alpha\}}$: $\lim_{x\to\alpha+0} F_1(x) = F_1(\alpha)$, $\lim_{x\to\alpha-0} F_1(x) = F_1(\alpha) - 2^{-m-\{\alpha\}}$. So for each point α the jump equals $F_2(\alpha)$. But F_2 is zero and continuous for all remaining points. Hence, the only difference between $F_1(x)$ and $F_1(x) - F_2(x)$ is that the latter function is continuous from the left, whereas F_1 is continuous from the right. Consequently, also the Fourier coefficients coincide, which entails in particular that the mean value of the linear term of $C_{w,bu}(N)$ is $c_0 = -0.7005738875\dots$.

The biggest jumps of $A^*(x) = F_1(x) - F_2(x)$ occur at integer points. Let α denote such a point. Then $\lim_{x\to\alpha-0} A^*(x) = A^*(\alpha) = -1$, whereas $\lim_{x\to\alpha+0} A^*(x) = 0$. Regarding the fluctuation about the main term this means that the worst case is "best" if N is a power of 2 and is "worst" if N has the form $N = 2^m + 1$.

2.3 The average Case

To get the average number of comparisons one only has to sum the expected values (4) over all merge processes. Unfortunately, the resulting expression is rather unwieldy.

$$\begin{aligned} C_{a,bu}(N) = &\sum_{0<j\le k} \frac{2^{2j-1}}{2^{j-1}+1}\Big\lfloor\frac{N}{2^j}\Big\rfloor \\ &+ \sum_{0<j\le k} b_j 2^j (b_{j-1}b_{j-2}\dots b_0)_2 \Big(\frac{1}{2^j+1} + \frac{1}{(b_{j-1}b_{j-2}\dots b_0)_2+1}\Big). \end{aligned} \tag{7}$$

The first sum is

$$\begin{aligned} \sum_{0<j\le k} \frac{2^{2j-1}}{2^{j-1}+1}\Big\lfloor\frac{N}{2^j}\Big\rfloor &= \sum_{0<j\le k} 2^j \Big\lfloor\frac{N}{2^j}\Big\rfloor - \sum_{0<j\le k} \frac{2^j}{2^{j-1}+1}\Big(\frac{N}{2^j} - \Big\{\frac{N}{2^j}\Big\}\Big) \\ &= \sum_{0<j\le k} 2^j \Big\lfloor\frac{N}{2^j}\Big\rfloor - N\sum_{0<j\le k} \frac{1}{2^{j-1}+1} + \sum_{0<j\le k} \frac{2^j}{2^{j-1}+1}\Big\{\frac{N}{2^j}\Big\} \end{aligned}$$

$$= \sum_{0<j\le k} 2^j \left\lfloor \frac{N}{2^j} \right\rfloor - N(\alpha' + O(2^{-k})) + O(\log N)$$

$$= \sum_{0<j\le k} 2^j \left\lfloor \frac{N}{2^j} \right\rfloor - \alpha' N + O(\log N), \tag{7a}$$

where the constant α' is well known, $\alpha' = \sum_{j\ge 0}(2^j+1)^{-1} = 1.26449\,97803\ldots$, and the error term is much smaller than linear. Furthermore,

$$\sum_{0<j\le k} \frac{b_j 2^j}{2^j+1}(b_{j-1}b_{j-2}\ldots b_0)_2$$

can be expressed as

$$\sum_{0<j\le k} \frac{b_j 2^j}{2^j+1} \cdot 2^j \left\{\frac{N}{2^j}\right\} = \sum_{0<j\le k} b_j 2^j \left\{\frac{N}{2^j}\right\} + O(\log N), \tag{7b}$$

where again the error term does not affect the linear term. Combining $(7a)$ and $(7b)$ we thus obtain

$$\begin{aligned}
&\sum_{0<j\le k} 2^j \left\lfloor \frac{N}{2^j} \right\rfloor - \alpha' N + \sum_{0<j\le k} b_j 2^j \left\{\frac{N}{2^j}\right\} + O(\log N) \\
=&\sum_{0<j\le k} 2^j \left(\frac{N}{2^j} - \left\{\frac{N}{2^j}\right\}\right) - \alpha' N + \sum_{0<j\le k} b_j 2^j \left\{\frac{N}{2^j}\right\} + O(\log N) \\
=&\lfloor \lg N \rfloor N - \alpha' N + \sum_{0<j\le k} (b_j - 1) 2^j \left\{\frac{N}{2^j}\right\} + O(\log N) \\
=&N \lg N + N\Big(\frac{1}{2^{\{\lg N\}}} \sum_{0<j\le k} (b_j - 1)\frac{(b_{j-1}\ldots b_0)_2}{2^{\lfloor \lg N \rfloor}} - \{\lg N\} - \alpha'\Big) + O(\log N) \\
=&N \lg N + N\Big(F_1(\lg N) - \alpha'\Big) + O(\log N)\,.
\end{aligned}$$

The remaining sum of (7) is

$$\sum_{0<j\le k} b_j 2^j \,\frac{(b_{j-1}b_{j-2}\ldots b_0)_2}{(b_{j-1}b_{j-2}\ldots b_0)_2+1}\,.$$

This expression also shows a quasi-periodical behavior. For larger N it can be approximated by the function $N(1 - 2^{v_2(N)}/N) = N(1 - F_2(\lg N))$. Accordingly, $C_{a,bu}(N)$ can be represented by

$$\begin{aligned}
&N \lg N + N\Big(F_1(\lg N) - F_2(\lg N) + 1 - \alpha'\Big) \\
=\;& N \lg N + N B^*(\lg N)\,,
\end{aligned} \tag{8}$$

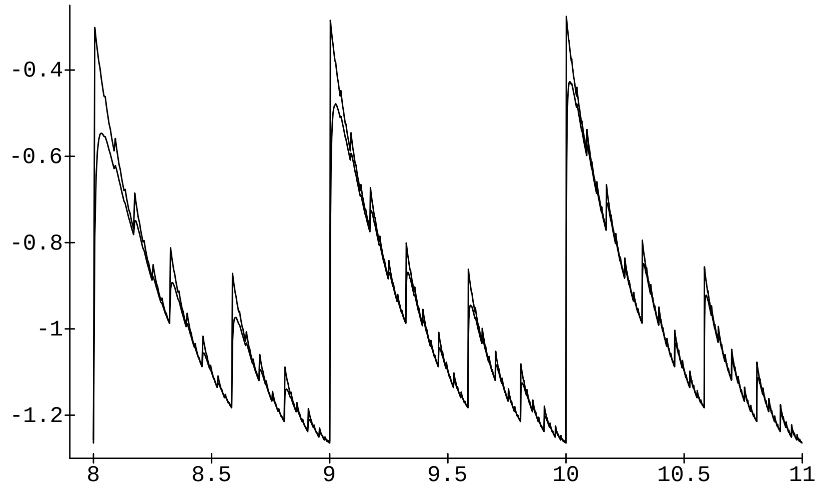

Fig. 3: The lower graph represents the true linear term for the average case, expressed by $(C_{a,bu}(N) - N \lg N)/N$. The values are plotted against $\lg N$ for $N = 2^8 .. 2^{11}$. The upper graph corresponds to the approximating function $B^*(x)$.

where $B^*(x) = A^*(x) + 1 - \alpha'$. Hence the mean value of $B^*(x)$ is $c_0 + 1 - \alpha' = -0.96507\,36678\ldots$. The minimum, realized at the integer points, is $-\alpha' = -1.2644997803\ldots$. The supremum is $\lim_{x\to\gamma+0} B^*(x) = 1-\alpha' = -0.2644997803\ldots$, where γ is integer. It is apparent from Fig. 3 that the limiting function $B^*(x)$ very well approximates the true behavior of the linear term of $C_{a,bu}$.

3. Top-Down versus Bottom-Up Mergesort

3.1 The best Case

Equation (5) shows that bottom-up mergesort in the best case behaves exactly like top-down mergesort regarding the number of comparisons, i.e. $C_{b,bu}(N) = C_{b,td}(N) = C_b(N)$. The behavior of $C_b(N)$ is, as we saw,

$$C_b(N) = \frac{1}{2} N \lg N + N \cdot F_0(\lg N),$$

where F_0 is a continuous, nowhere differentiable function which is periodic with period 1. F_0 is explicitly defined in section 2.1, where also its Fourier expansion is given. The mean value of F_0 is $-0.145599455\ldots$, its maximum is 0 and its minimum is $(\lg 3)/2 - 1 = -0.207518749\ldots$. The maximum is attained in every integer point and the minimum is realized in the points congruent to $\lg(4/3) = 0.415037499\ldots$.

3.2 The worst Case

The following considerations show that the top-down variant behaves somewhat better in the worst case than the bottom-up variant, which is well known to require

$$C_{w,td}(N) = (k+1)N - 2^{k+1} + 1$$

comparisons in the worst case (cf. e.g. [Gonnet]). Taking the difference one gets

$$C_{w,bu}(N) - C_{w,td}(N) = 2^{k+1} - N - \sum_{0<j\le k} (1-b_j)(b_{j-1}b_{j-2}\ldots b_0)_2 - 2^{v_2(N)}.$$

But

$$2^{k+1} - N = \big((1-b_k)(1-b_{k-1})\ldots(1-b_1)(1-b_0)\big)_2 + 1,$$

and the above difference may consequently be written as

$$\sum_{0\le j\le k} (1-b_j)2^j - \sum_{0<j\le k} (1-b_j)(b_{j-1}b_{j-2}\ldots b_0)_2 - 2^{v_2(N)} + 1.$$

Let $v_2(N) = r$. Of course $0 \le r \le k$ and $b_r = 1$. Dissecting both sums with regard to r we finally obtain

$$C_{w,bu}(N) - C_{w,td}(N) = \sum_{r<j\le k} (1-b_j)[2^j - (b_{j-1}b_{j-2}\ldots b_0)_2] \ge 0.$$

Hence the worst-case number of comparisons for both merging strategies is equal only for N's with a binary representation of the form $N = (1^+0^*)_2$. In general the bottom-up algorithm performs more comparisons in the worst case. The maximum for $N \in [2^k, 2^{k+1}]$ occurs for $N = 2^k + 1$ sorting elements and amounts to $2^k - k + 1$ additional comparisons.

Incidentally, the more favorable behavior of the top-down variant in the worst case is not so surprising because of the close connection between the top-down splitting strategy and Fano coding, from which we may conclude that the top-down algorithm has the best worst-case behavior (is *minimax-optimal*, cf. [Knuth, p.193]) among all possible splitting strategies.

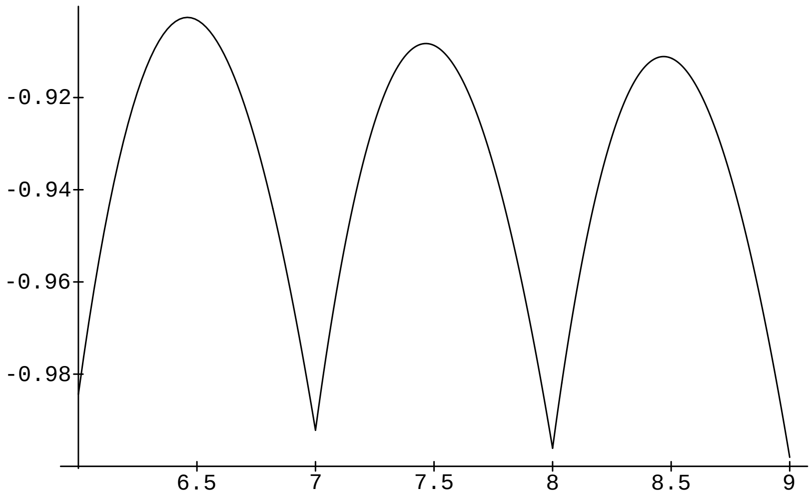

Fig. 4: The fluctuation of the low order terms for the worst case, expressed by $(C_{w,td}(N) - N \lg N)/N$. The values are plotted against $\lg N$ for $N = 2^6 .. 2^9$.

Fig.4 shows the fluctuation of the low order terms for the top-down variant. As already mentioned in the introduction, the worst case behavior of top-down mergesort can be described by (cf. [Flajolet, Golin])

$$C_{w,td}(N) = N \lg N + NA(\lg N) + 1,$$

where

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

The mean value of $A(x)$ is $-0.9426950408\ldots$. The extreme values are -1 and $-(1 + \log\log 2)/\log 2 = -0.91392\ldots$, where the minimum is attained at integer points and the maximum at the points congruent to $1 + \log\log 2/\log 2$.

On the other hand it has been shown in section 2.2 that

$$C_{w,bu}(N) = N \lg N + NA^*(\lg N) + 1,$$

where A^* is periodic with period 1. A^* is continuous from the left with jumps of $2^{-m-\{\alpha\}}$ in the points α for which $2^{\{\alpha\}} = (1.\beta_1 \ldots \beta_m)_2$ is dyadic. For the remaining points A^* is continuous. A^* is nowhere differentiable. The mean value is $-0.7005738875\ldots$. The minimum is -1 and it is attained in the integer points. The supremum is $\lim_{x\to\gamma+0} A^*(x) = 0$, where γ is an integer. Additionally, as a consequence of the considerations at the begining of the present section, we also have $A(x) \le A^*(x)$, where equality holds for the points congruent to $\lg(2^{k+1}-1)-k$, $k = 0, 1, 2, \ldots$. On the average the difference is about 0.24. Fig.4a shows a plot of the functions $A(x)$ and $A^*(x)$.

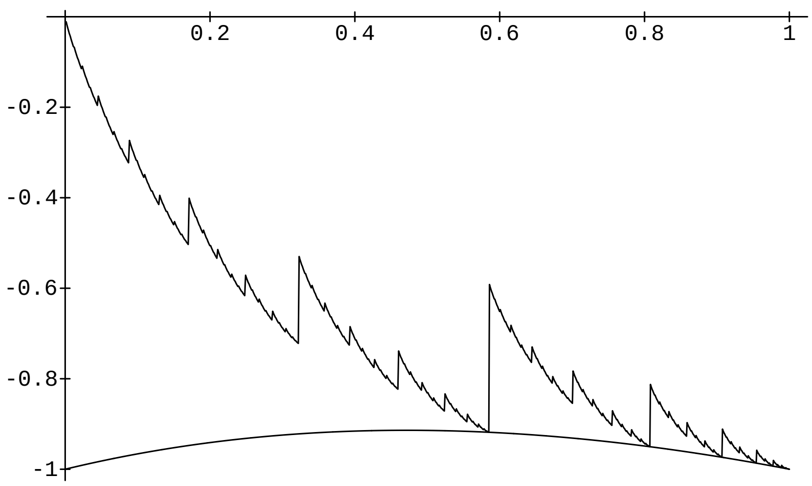

Fig.4a: The functions $A(x)$ and $A^*(x)$ for $x \in (0, 1]$.

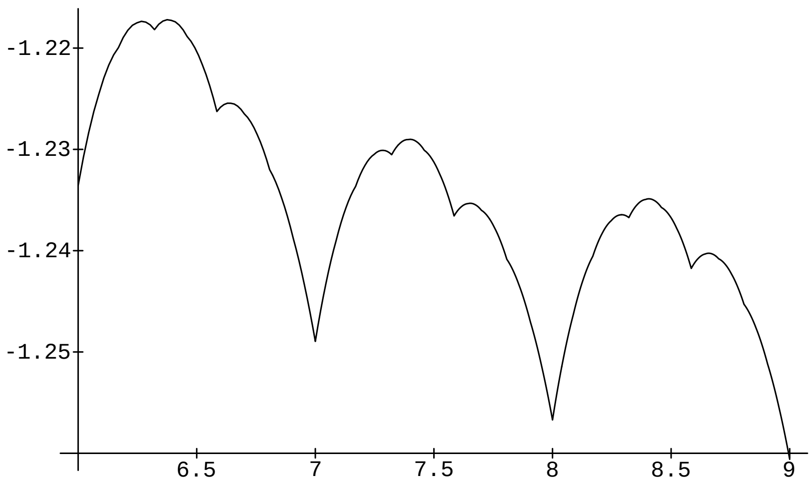

Fig. 5: The fluctuation of the low order terms for the average case, expressed by $(C_{a,td}(N) - N \lg N)/N$. The values are plotted against $\lg N$ for $N = 2^6 .. 2^9$.

3.3 The average Case

The average case number of comparisons for the top-down algorithm is

$$C_{a,td}(N) = (k+1)N - 2(2^{k+1} - 1) + 2 \sum_{0 \le j \le k} \frac{2^j}{\lfloor \frac{N}{2^{j+1}} \rfloor + 2}$$
$$+2 \sum_{0 \le j \le k} \frac{(1 - b_j)(2^j - (b_{j-1} \dots b_1 b_0)_2)}{\left(\lfloor \frac{N}{2^{j+1}} \rfloor + 1\right)\left(\lfloor \frac{N}{2^{j+1}} \rfloor + 2\right)}. \tag{9}$$

Flajolet and *Golin* recently have studied the periodical fluctuation of $C_{a,td}(N)$ as shown in Fig. 5. They have shown that

$$C_{a,td}(N) = N \lg N + N B(\lg N) + O(N^\epsilon), \quad \epsilon > 0,$$

where $B(x)$ is a continuous non-differentiable periodic function with period 1 that has an explicit Fourier expansion $B(x) = \sum_{k=0,\pm1,\pm2,\dots} b_k \exp\{2\pi i k x\}$. The mean value b_0 of $B(x)$ equals

$$\frac{1}{2} - \frac{1}{\log 2} - \frac{1}{\log 2} \sum_{j \ge 1} \frac{2}{(j+1)(j+2)} \log\left(\frac{2j+1}{2j}\right) = -1.24815\,20420\,99653\,88489\dots.$$

The remaining Fourier coefficients are

$$b_k = \frac{1}{\log 2} \frac{1 + \Psi(\chi_k)}{\chi_k(\chi_k + 1)} \qquad \text{where again} \quad \chi_k = \frac{2\pi i k}{\log 2}$$

and

$$\Psi(s) = \sum_{j \ge 0} \frac{2}{(j+1)(j+2)} \left[\frac{-1}{(2j)^s} + \frac{1}{(2j+1)^s}\right].$$

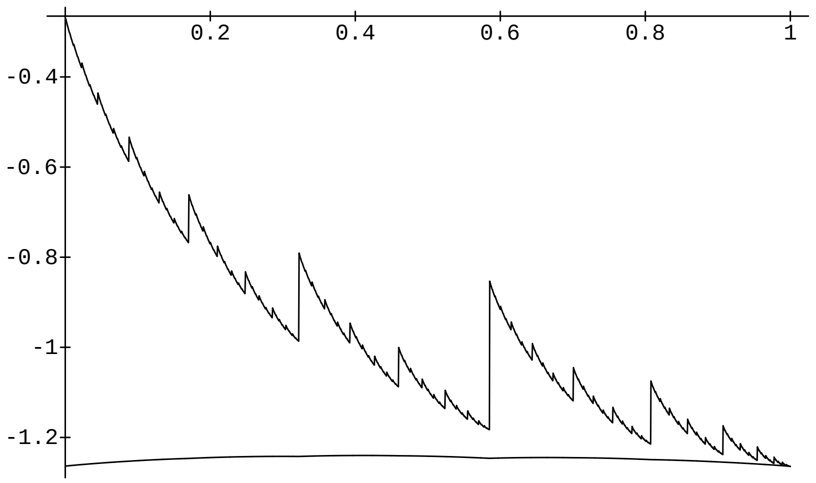

Fig. 5a: The functions $B(x)$ and $B^*(x)$ for $x \in (0,1]$.

The minimum of $B(x)$ is $-\sum_{j\geq 0}(2^j+1)^{-1} = -\alpha' = -1.26449\,97803\ldots$ and occurs at integer points. The maximum is $-1.24075\,0572 \pm 10^{-9}$ and is attained in the points congruent $0.41281\,3929\ldots$.

Comparing $B(x)$ with $B^*(x)$ shows that on the average their difference is about 0.28. Both functions coincide at integer points, where they both realize their minimum. It seems that the relation $B(x) \leq B^*(x)$ holds for all points (compare also Fig. 5a). We even believe that the top-down splitting strategy has the best average-case behavior (is *minimean-optimal*, cf. [Knuth, p. 193]) among all possible splitting strategies.

4. Concluding Remarks

Our results together with the results of *Flajolet* and *Golin* show that the average-case number of comparisons for both variants differ at most by N. The mean difference is only about $0.28N$.

	best case	average case	worst case
top-down	$\frac{1}{2}N\lg N - 0.146N$	$N\lg N - 1.248N$	$N\lg N - 0.943N$
bottom-up	$\frac{1}{2}N\lg N - 0.146N$	$N\lg N - 0.965N$	$N\lg N - 0.701N$

Tab. 1: Comparison of top-down and bottom-up mergesort. For the linear term the mean value is shown.

This slightly better performance of the top-down algorithm is contrasted by an important restriction, namely that the number of elements to be sorted must be known in advance. This seems to be a serious shortcoming for a sorting algorithm whose natural application field is to operate on *lists*. In the literature it is recommended to overestimate the number of elements in the list, if it is unknown. It seems very likely that the narrow time-advantage of the top-down variant gets lost in that case.

5. References

Chéno, L. (1981) *Profils limites d'histoires sur les dictionaires et les fils de priorité; Application aux files binomiales*, Thèse, Université de Paris-Sud.

Delange, H. (1975) Sur la fonction sommatoire de la fonction "Somme des Chiffres". *Enseignement Math.(2)*, **21**, 31–47.

Flajolet, P., Grabner, P., Kirschenhofer, P., Prodinger, P., Tichy, R., F. (1994) Mellin transforms and asymptotics: digital sums. *Theoretical Computer Science*, **123**, 291–314.

Flajolet, P., Golin, M. (1992) Mellin transforms and asymptotics: the mergesort recurrence. *INRIA research report*, **1612**. Accepted for publication in *Acta Informatica*.

Foster, D. M. (1987) Estimates for a remainder term associated with the sum of digits function. *Glasgow Math. J.*, **29**, 109–129.

Golin, M., Sedgewick, R. (1993) Queue-mergesort. *Inf. Process. Lett.*, **48**, 253–259.

Gonnet, G. H., Baeza-Yates, R. (1991) *Handbook of Algorithms and Data Structures*, 2nd ed., Addison-Wesley, Reading, MA.

Knuth, D. E. (1973) *The Art of Computer Programming: Sorting and Searching*, Addison-Wesley, Reading, MA.

Panny, W. (1988) Straight two-way mergesort: der Algorithmus und seine Analyse, in *Statistik, Informatik und Ökonomie — Josef Roppert zum 60. Geburtstag*, (ed.: W.H. Janko), 216–236, Springer, Berlin.