An alternative approach to the efficiency of recursive merge sort*

Tibor Ásványi[a]

[a]Eötvös Loránd University, Faculty of Informatics
asvanyi@inf.elte.hu

Abstract

The operational complexity $\Theta(n \lg n)$ of merge sort is usually calculated by solving the recurrence

$$T(n) = \begin{cases} T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) & \text{if } n > 1, \\ \Theta(1) & \text{if } n = 1, \end{cases}$$

where n is the length of the sequence of keys to be sorted, $T(n)$ is either the best-case or the worst-case asymptotic running time of the algorithm, $\Theta(n)$ is that of division + sorted merge, and $\Theta(1)$ is that of the base case [1].

In this paper we invent an alternative approach: We analyze the structure of the tree of recursive calls, consider its depth, and estimate the number of steps [2] of computation at the different levels of that tree. Compared to the equation above we use a more strict notation [3] and argue about its scientific and didactic advantages in efficiency analysis of algorithms in general.

Keywords: algorithm, merge sort, recursion, operational complexity, asymptotic running time, efficiency analysis, education

MSC: 68P05, 68P10, 68P20, 68Q25

References

[1] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., Introduction to Algorithms (Third Edition), *The MIT Press* (2009).

[2] Weiss, Mark Allen, Data Structures and Algorithm Analysis, *Addison-Wesley* (2013).

[3] Neapolitan, Richard E. Foundations of Algorithms (Fifth Edition), *Jones & Bartlett Learning* (2015).

*Thanks to the Eötvös Loránd University, Faculty of Informatics for financial support.