

HOMework 2

ALGORITHMS AND DATA STRUCTURES (CH08-320201)

Spring 2018

Prof. Dr. Michael Sedlmair
Computer Science & Electrical Engineering
Jacobs University

Due on Monday, Feb 19, 2018, 23:55.

Problem 1: Merge Sort

(7+3+4+2=16 points)

- (a) Implement a variant of Merge Sort that does not divide the problem all the way down to subproblems of size 1. Instead, when reaching subsequences of length k it applies Insertion Sort on these n/k subsequences.
- (b) Apply it to the different sequences from Problem 2 (from last homework) for different numbers of k . Add the computation times to the plots you had generated in Problem 2.
- (c) How do the different values of k change the best-, average-, and worst-case asymptotic time complexities for this variant? Explain/proove your answer.
- (d) Based on the results from (b) and (c), how would you choose k in practice? Briefly explain.

Problem 2: Recurrences

(2+2+2+2+2=10 points)

Use substitution method, recursion tree, or master method to derive upper and lower bounds for $T(n)$ in each of the following recurrences. Make the bounds as tight as possible. Assume that $T(n)$ is constant for $n \leq 2$.

- (a) $T(n) = 36T(n/6) + 2n$.
- (b) $T(n) = 5T(n/3) + 17n^{1.2}$.
- (c) $T(n) = 12T(n/2) + n^2 \lg n$.
- (d) $T(n) = 3T(n/5) + T(n/2) + 2^n$.
- (e) $T(n) = T(2n/5) + T(3n/5) + \Theta(n)$.

Remarks

Solutions have to be handed in via Moodle by the due date. For late submissions you need to get in contact with the TAs directly. You need to upload one zip-file that contains a PDF-file for the theoretical parts and source files (no executables or object files) for the programming assignments. The source files need to include a makefile. Programming assignments need to be handed in as C, C++, or Python code. **Please write your own code.** It is ok to take snippets from online resources, but they need to be clearly marked.