

HOMework 1

ALGORITHMS AND DATA STRUCTURES (CH08-320201)

Spring 2018

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

Due on Monday, Feb 12, 2018, 23:59.

Problem 1: Asymptotic Analysis

(2+2+2+2=8 points)

Considering the following pairs of functions f and g , show for each pair whether $f \in \Theta(g)$, $f \in O(g)$, $f \in o(g)$, $f \in \Omega(g)$, $f \in \omega(g)$, $g \in \Theta(f)$, $g \in O(f)$, $g \in o(f)$, $g \in \Omega(f)$, or $g \in \omega(f)$.

- (a) $f(n) = 3n$ and $g(n) = n^3$.
- (b) $f(n) = 7n^{0.7} + 2n^{0.2} + 13 \log n$ and $g(n) = \sqrt{n}$.
- (c) $f(n) = n^2 / \log n$ and $g(n) = n \log n$.
- (d) $f(n) = (\log(3n))^3$ and $g(n) = 9 \log n$.

Problem 2: Selection Sort

(2+5+3+5+1=16 points)

In class, we discussed Insertion Sort. Selection Sort is similar to Insertion Sort and works as follows. Given an array of elements, you always take the current element and exchange it with the smallest element that can be found on the right hand side of the current element. In doing so, you will gradually build up a sorted list on the left side (just as in Insertion Sort), and in each iteration “attach” the smallest element from the remaining unsorted right side to it.

- (a) *Implement* Selection Sort.
- (b) Show that Selection Sort is correct (consider the loop invariant).
- (c) Generate random input sequences of length n as well as sequences of length n that represent the worst case and the best case for the Selection Sort algorithm. Briefly describe how you generated the sequences (e.g. programmatically with a random sequence generator).
- (d) Run the algorithm on sequences of (c) with length n for increasing values of n and measure the computation times. Plot curves that show the computation time of the algorithm in the best, average, and worst case for increasing input length n . Note that in order to compute reliable measurements for the average case, you have to run the algorithm multiple times for each entry in your plot. You can use whatever plotting method you like (R, Matlab, Excel, by hand, ...).
- (e) Interpret the plots in (d) with respect to asymptotic behavior and constants.

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.