University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithms and Datastructures
Exercise Sheet 2

Exercise 1: $\mathcal{O}$-notation *(9 Points)*

Prove or disprove the following statements. Use the *set definition* of the $\mathcal{O}$-notation (lecture slides week 2, slides 11 and 12).

(a) $2n^3 + 4n^2 + 7\sqrt{n} \in \mathcal{O}(n^3)$ *(1 Point)*

(b) $n \cdot \log_3(n) \in \omega(n \cdot \log_5(n))$ *(2 Points)*

(c) $2^n \in o(n!)$ *(2 Points)*

(d) $2\log_2\left(n^2\right) \in \Omega((\log_2 n)^2)$ *(2 Points)*

(e) $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$ for non-negative functions f and g. *(2 Points)*

Exercise 2: Sorting by asymptotic growth *(4 Points)*

Sort the following functions by their asymptotic growth. Write $g <_{\mathcal{O}} f$ if $g \in \mathcal{O}(f)$ *and* $f \notin \mathcal{O}(g)$. Write $g =_{\mathcal{O}} f$ if $f \in \mathcal{O}(g)$ *and* $g \in \mathcal{O}(f)$ (no proof needed).

$\sqrt{n}$	2^n	$n!$	$\log(n^3)$
3^n	n^{100}	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$(n+1)!$	$n \log n$
$2^{(n^2)}$	n^n	$\sqrt{\log n}$	$(2^n)^2$

Exercise 3: Event Scheduling *(7 Points)*

There are n events $e_1, \ldots, e_n$, each described by their starting time $s_1, \ldots, s_n$ and their ending times $t_1, \ldots, t_n$. You can only attend one event at a time and if you started an event, you must attend the entire event. We ignore traveltimes between events. The goal is to attend as many events as possible.

Devise an Algorithm that computes the maximum number of events you can attend. Why is your algorithm correct? What is the asymptotic complexity of the algorithm? Can you devise an algorithm that is $o(n^2)$?