Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Algorithmics 1
WS 2018/19

Exercise 1

Task 1
Prove or disprove the following statements:

(a) $n \log n \in \mathcal{O}(n^2)$

(b) $n^2 \in \mathcal{O}(n)$

(c) $n^n \in \Omega(2^n)$

(d) $n - \log n \in o(n)$

(e) $2 + (-1)^n \in \Theta(1)$

(f) $n! \in \omega(2^n)$

Task 2
Let $f\colon \mathbb{N} \to \mathbb{N}$ with $f(n) \in \Theta(n)$. Prove or disprove the following statements:

(a) $f(n)^k \in \Theta(n^k)$ for all $k \in \mathbb{N}$, $k \geq 1$

(b) $2^{f(n)} \in \Theta(2^n)$

Task 3
Use the Master Theorem to determine the asymptotic growth of the following functions:

(a) $T_1(n) = 7 \cdot T_1\left(\frac{n}{2}\right) + 4n$

(b) $T_2(n) = 7 \cdot T_2\left(\frac{n}{2}\right) + 1000n^2$

(c) $T_3(n) = 8 \cdot T_3\left(\frac{n}{2}\right) + n^2$

(d) $T_4(n) = 8 \cdot T_4\left(\frac{n}{2}\right) + n^3$

(e) $T_5(n) = 6 \cdot T_5\left(\frac{n}{3}\right) + n^3$

(f) $T_6(n) = 64 \cdot T_6\left(\frac{n}{8}\right) + n^2$

Task 4
The algorithm *Insertion Sort* sorts an array $a_1, \ldots, a_m$ in the following way: From $i = 0$ to $i = m - 1$ it assumes that $a_1, \ldots, a_i$ has already been sorted. It then sorts $a_1, \ldots, a_{i+1}$ by putting a_{i+1} in the correct position, comparing a_{i+1} to every element of $a_1, \ldots, a_i$.

(a) Give the recurrence equation for the function $T(n)$ which is the number of comparisons needed for an input array of n elements.

(b) Why can you not apply the Master Theorem to solve the recurrence equation?

(c) Solve the recurrence equation manually.

(d) Give a function $f\colon \mathbb{N} \to \mathbb{N}$ of the form $f(n) = n^k$ for some $k \in \mathbb{N}$ such that $T \in \Theta(f)$.