Martin Ziegler
Carsten Rösnick

Distributed on 18-Oct-2011
Solutions to be submitted by 25-Oct-2011

# Complexity Theory

## WS 2011/2012, Exercise Sheet #0

**AUFGABE 1:**

a) Compare the following three classes of asymptotic growth (Landau's notation): which one includes which one; which coincide? Justify your claims!

i) $\mathcal{O}(2^n)$ ii) $2^{\mathcal{O}(n)}$ iii) $\mathcal{O}\big(2^{\mathcal{O}(n)}\big)$

b) Does $n^{\log(n)}$ mean polynomial oder exponential growth? Why?

c) Simplify: $n^{\log\log(n)/\log(n)}$. Prove: $a^{\log b} = b^{\log a}$.

d) Verify: $\max(f,g) = \Theta(f+g)$ and $n! = 2^{\Theta(n \log n)}$. Hint: Stirling.

e) Determine the asymptotic growth of $n \mapsto \binom{n}{K}$ and of $n \mapsto \sum_{k=0}^{K} \binom{n}{k}$.
Here $K$ is considered constant. What can happen in case $K$ depends on $n$?

f) Consider the recursive equation $f(n) = a \cdot f\big(\lceil n/2 \rceil\big) + b \cdot n^c$ ($a > 2^c$, $b, c > 0$, $n \geq 2$). Show that it defines a monotonically nondecreasing function $f : \mathbb{N} \to \mathbb{R}$ and determine its asymptotic growth.

g) Prove or disprove: For every function $f : \mathbb{N} \to \mathbb{N}$, it holds $f(n) = \mathcal{O}(n)$ or $f(n) = \Omega(n)$.

**AUFGABE 2:**

a) Recall the high school method for long addition. Describe semi-informally an algorithm for adding two integers in binary, i.e. on input $(a_{n-1}, \ldots, a_0)$ and $(b_{n-1}, \ldots, b_0)$ over $\{0,1\}$. Specify the (format of the) output and the basic operations employed. Determine the asymptotic running time. Is it optimal?

b) Recall the high school method for long multiplication. Describe an algorithm for multiplying two integers of length $n$ in binary and analyze its running time.

c) Describe semi-informally an algorithm for multiplying an integer of length $n$ in binary with the number $2^m$ and determine its running time.

d) $(aX+b) \cdot (cX+d) \; = \; rX^2 + (s-r-t)X + t, \quad r := a \cdot c,\; t := b \cdot d,\; s := (a+b) \cdot (c+d)$.
Describe an algorithm for multiplying two integers of length $n$ in subquadratic running time, thus improving over b).

e) Describe an algorithm performing long division with remainder of two integers of length $n$ in binary within running time $\mathcal{O}(n^2)$.

f) Describe an algorithm deciding whether a given $n$-bit integer is prime or not. What running time do you achieve?
Note: In 2003 an algorithm was presented that achieves the same in running time $\mathcal{O}(n^{12})$.

---

*Logarithms are always to be understood with respect to basis 2.
To have this course re-scheduled according to your constraints, please vote at `doodle.de/h4yy2mf85keva852`