Tutorial 1: Basics

Tutorial 1 will take place in week 2. You should prepare solutions, but you don't have to hand them in and they won't get marked.

Exercise 1 *Induction Proofs*

Recall the principle of doing proofs by mathematical induction.

1. Prove by mathematical induction that $n! \geq \text{fib}(n)$ for all $n \geq 0$. Note that $fib(n)$ denotes the nth Fibonnaci number.
2. Let a and $r \neq 1$ be real numbers. Prove by mathematical induction the geometric series, i. e. that
$$\sum_{i=0}^{n} a \cdot r^i = \frac{a(1 - r^{(n+1)})}{1 - r}$$
holds for all natural numbers n.

Exercise 2 *Complexity Notation*

Solve Exercise 2.1 in the book of Mehlhorn/Sanders (page 22).

Exercise 3 *Complexity Notation*

Solve Exercise 2.2 in the book of Mehlhorn/Sanders (page 23).

Exercise 4 *Complexity Notation*

Solve Exercise 2.3 and 2.4 in the book of Mehlhorn/Sanders (page 23).

Exercise 5 *Complexity Notation*

Is it true that if $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, then $h(n) = \Theta(f(n))$?

Exercise 6 *Complexity Notation*

Is it true that if $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $h(n) = \Omega(f(n))$?

Exercise 7 *Complexity Notation*

Is it true that a $\Theta(n^2)$ algorithm always takes longer to run than a $\Theta(logn)$ algorithm?

Exercise 8 *Complexity Notation*

For each pair of functions given below, point out the asymptotic relationships that apply: $f = O(g)$, $f = \Theta(g)$, $f = \Omega(g)$.

- $f(n) = \sqrt{n}$ and $g(n) = log(n)$
- $f(n) = 1$ and $g(n) = 2$
- $f(n) = 1000 \cdot 2^n$ and $g(n) = 3^n$
- $f(n) = 4^{n+4}$ and $g(n) = 2^{2n+2}$
- $f(n) = 5nlog(n)$ and $g(n) = nlog(5n)$
- $f(n) = n!$ and $g(n) = (n+1)!$

Exercise 9 *Complexity Notation*

Prove that $n^k = o(c^n)$ for any integer k and any $c > 1$.