**Lab 3**

**David Blumenthal, Florian Hofer,
Werner Nutt, Daniele Porello**

# 3. Asymptotic Complexity

### Comparison According to Asymptotic Complexity

Below you find a list of functions that could appear as functions describing the running time of algorithms.

1. $50 \cdot \log_2 n$
2. $5 \cdot n + n^2 + 1$
3. $(\log_{10} n)^2$
4. $2^n + 5$
5. $3^{\sqrt{n}}$
6. $5^3 \cdot n$
7. $3 \cdot \log_{10} n$
8. $(n+1)!$
9. $4^{\log_2 n}$
10. $\sqrt{n}$.

Order the following functions according to their asymptotic complexity. Start with the function having the smallest asymptotic complexity and move on to the function having the next largest one. That is, write them as a sequence $f_1, f_2, \ldots, f_9$ such that $f_1(n) = O(f_2(n))$, $f_2(n) = O(f_3(n))$, etc.
Indicate as well functions $f$, $g$ that are asymptotically equivalent, that is, where both $f(n) = O(g(n))$ and $g(n) = O(f(n))$ hold.
Find out as well the cases where $f(n) = O(g(n))$, but the converse $g(n) = O(f(n))$ does not hold.

**Hint:** Consult the Toolbox for Asymptotic Analysis on the lab page of the course.