Bioinformatics Group
Prof. Dr. Rolf Backofen
Florian Eggenhofer
Michael Uhl
Rick Gelhausen

Entwurf, Analyse und Umsetzung von Algorithmen

Exercise sheet 3

Exercise 1 (6 points)
Show that $\log_2 n = \mathcal{O}(n)$ holds. Directly use the definition of $\mathcal{O}$ by determining n_0 and C such that $\log_2 n \leq C \cdot n$ holds for all $n \geq n_0$.

Show that $\log_2 n = \Omega(n)$ does not hold. Directly use the definition of Ω by showing that for each given $C > 0$ and n_0 there exists a $n \geq n_0$ which violates the definition of Ω (i.e. that $\log_2 n \leq C \cdot n$). Consider also that C can be smaller than 1.

Exercise 2 (7 points)
Argue that the propositions in Exercise 1 do not only hold for $\log_2$, but in the general case $\log_b$ for any single given $b > 1$ that does not depend on n. Remark: That is why in runtime analyses you often find log written without the base. Why is it important that $b > 1$? Describe what happens if $b = 1$ or $b < 1$. Why is it important that b does not depend on n? Give an example where b does depend on n and one of the propositions above does not hold anymore (you can choose which one you want to use).

Exercise 3 (7 points)
Order the following functions f_1, f_2, f_3, f_4, f_5 according to their runtime complexity such that $f_i = \mathcal{O}(f_{i+1})$ holds for $i = 1, 2, 3, 4$. Also determine for which i $f_i = \Theta(f_{i+1})$ holds and for which not. Justify your decisions, particularly for the i cases where $f_i = \Theta(f_{i+1})$ does not hold. You can use the limit definition of $\mathcal{O}$ and Θ for all your justifications.

$$n^2$$
$$n \log_{10} n$$
$$n^2 \log_2(n^2)$$
$$\sqrt{n}$$
$$n \log_2(n^2)$$