Exercises Complexity Theory

Lecture 2

April 17, 2023

Only the exercises where points are given can be handed in.
(The maximum number of points per exercise is written in the margin.)

To be handed in on **April 24, 2023**, in Brightspace under Assignment 1, **deadline: 10:00 AM.**

Exercise 1. An $n \times m$ chocolate bar can be broken into unit squares, by breaking along the unit lines. Claim: the number of breaks required is $nm - 1$.
Prove this by strong induction

Exercise 2. We prove a number of standard useful properties about log. Let $a, b, c > 0$.

(a) Prove that we have $\log_a b \cdot \log_b c = \log_a c$.
(Hint: consider a^x for x the two sides of the equation)

(b) From (a), show that one can "change log-base at a constant cost factor", by showing that $\log_b f(n) = d \log_a f(n)$ for some constant $d > 0$.

(c) From (a), show that $a^{\log_c b} = b^{\log_c a}$
(Hint: take $\log_a$ at both sides of this equation.)

(d) Given $d > 0$, prove that $\log x + d > \log(x + d)$ for sufficiently large x. When is x "sufficiently large"?

(20) **Exercise** 3. We have a recursive algorithm whose time complexity $T(n)$ satisfies

$$T(n) = 7T(n-2) + 5T(n-3) + f(n),$$

with $f(n) = \Theta(n^3)$. Find the smallest integer k for which $T(n) = \mathcal{O}(k^n)$ and prove that $T(n) = \mathcal{O}(k^n)$. (Also argue why your k is the smallest.)

Exercise 4. Consider the algorithm for computing the median of an array, as it has been described and discussed in the lecture. We adapt the algorithm by splitting the input array A randomly in $\frac{n}{7}$ groups of 7 elements.

(10) (a) Give the adapted algorithm and give an equation for $T(n)$, the time the algorithm takes on an array of length n.

(20) (b) Analyze the complexity of the algorithm by giving the recursion tree and derive an upperbound $f(n)$ to the running time $T(n)$, where n is the length of A. Is the algorithm still linear?

(10) (c) Prove that the adapted algorithm is $\mathcal{O}(f(n))$, where $f(n)$ is the function you've derived from the recursion tree in (b).

Exercise 5. In this exercises, use a recursion tree to give a good asymptotic upperbound to $T(n)$. Then verify your bound using the substitution method.

(20) (a) Let $T(n)$ be given by

$$T(n) = 3T(n-2) + d$$

for some constant d.

(20) (b) Let $T(n)$ be given by

$$T(n) = 2T(\frac{n}{3} + 3) + n^2.$$