MAST31213 Complexity theory
Master's Programme in Mathematics and Statistics
Fall 2019
Exercise set 1

Read chapter 0 of the book (or look up the needed definitions from some other material).

Exercise 1. Prove that for fixed constants c, k,

$$n^k = o(c^n).$$

(You can consult your favourite Calculus book, if you don't remember how to prove this.)

Exercise 2. Prove that for any fixed k,

$$n^k = o(n^{k+1}).$$

Exercise 3. For each of the following pairs of functions f, g, determine whether $f = o(g)$, $g = o(f)$ or $f = \Theta(g)$. If $f = o(g)$ then find the first number n such that $f(n) < g(n)$:

(a) $f(n) = n^2$, $g(n) = 2n^2 + 100\sqrt{n}$.
(b) $f(n) = n^{100}$, $g(n) = 2^{n/100}$.
(c) $f(n) = n^{100}$, $g(n) = 2^{n^{1/100}}$.
(d) $f(n) = \sqrt{n}$, $g(n) = 2\sqrt{\log n}$.

Exercise 4. For each of the following recursively defined functions f, find a closed (nonrecursive) expression for a function g such that $f(n) = \Theta(g(n))$, and prove that this is the case. *Note*: You can assume that $f(1) = f(2) = \cdots = f(10) = 1$ and the recursive rule is applied for $n > 10$; the base case won't make any difference to the answer. Why is that?

(a) $f(n) = f(n-1) + 10$.
(b) $f(n) = f(n-1) + n$.
(c) $f(n) = 2f(n/2) + n$.
(d) $f(n) = 2f(n/2) + O(n^2)$.

Exercise 5. Complexity theory mainly studies *decision problems*, with yes- or no-answers. However, many interesting problems are not a priori of this form. How would you formulate factorisation of integers as a decision problem?