Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement of Computer Science
Johannes Lengler, Markus Püschel, David Steurer
Kasper Lindberg, Kostas Lakis, Lucas Pesenti, Manuel Wiedmer

29 September 2025

Algorithms & Data Structures — Exercise sheet 2 — HS 25

The solutions for this sheet are submitted on Moodle until 5 October 2025, 23:59.

Exercises that are marked by * are challenge exercises. They do not count towards bonus points.

You can use results from previous parts without solving those parts.

Exercise 2.1 *Induction.*

(a) Prove via mathematical induction that for all integers $n \geq 5$,

$$2^n > n^2 .$$

In your solution, you should address the base case, the induction hypothesis and the induction step.

(b) Let x be any real number. Prove via mathematical induction that for every positive integer n, we have

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i ,$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} .$$

Recall that the factorial of a positive integer n is defined as $n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$. We use a standard convention $0! = 1$, so $\binom{n}{0} = \binom{n}{n} = 1$ for every positive integer n.

In your solution, you should address the base case, the induction hypothesis and the induction step.

Hint: You can use the following fact without proof: for every $1 \leq i \leq n$,

$$\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i} .$$

Exercise 2.2 *Fibonacci numbers* **(1 point)**.

There are a lot of neat properties of the Fibonacci numbers that can be proved by induction. Recall that the Fibonacci numbers are defined by $f_0 = 0, f_1 = 1$ and the recursion relation $f_{n+1} = f_n + f_{n-1}$ for all $n \geq 1$. For example, $f_2 = 1, f_5 = 5, f_{10} = 55, f_{15} = 610$.

Prove that $f_n \geq \frac{1}{3} \cdot 1.5^n$ for $n \geq 1$.

In your solution, you should address the base case, the induction hypothesis and the induction step.

Asymptotic Notation

When we estimate the number of elementary operations executed by algorithms, it is often useful to ignore constant factors and instead use the following kind of asymptotic notation, also called O-Notation. We denote by $\mathbb{R}^+$ the set of all (strictly) positive real numbers and by $\mathbb{N}$ the set of all (strictly) positive integers. Let N be a set of possible inputs to our algorithm such that $f : N \to \mathbb{R}^+$ represents its runtime given an input.

Definition 1 (O-Notation)**.** For $f : N \to \mathbb{R}^+$,

$$O(f) \coloneqq \{g : N \to \mathbb{R}^+ \mid \exists C > 0\, \forall n \in N\; g(n) \le C \cdot f(n)\}.$$

We write $f \le O(g)$ to denote $f \in O(g)$. Some textbooks use here the notation $f = O(g)$. We believe the notation $f \le O(g)$ helps to avoid some common pitfalls in the context of asymptotic notation.

Instead of working with this definition directly, it is often easier to use limits in the way provided by the following theorem.

Theorem 1. *Let N be an infinite subset of $\mathbb{N}$ and $f : N \to \mathbb{R}^+$ and $g : N \to \mathbb{R}^+$.*

- *If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, then $f \le O(g)$ and $g \not\le O(f)$.*
- *If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C \in \mathbb{R}^+$, then $f \le O(g)$ and $g \le O(f)$.*
- *If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, then $f \not\le O(g)$ and $g \le O(f)$.*

The following theorem can also be helpful when working with O-notation.

Theorem 2. *Let $f, g, h : N \to \mathbb{R}^+$. If $f \le O(h)$ and $g \le O(h)$, then*

1. *For every constant $c > 0$, $c \cdot f \le O(h)$.*
2. *$f + g \le O(h)$.*

Notice that for all real numbers $a, b > 1$, $\log_a n = \log_a b \cdot \log_b n$ (where $\log_a b$ is a positive constant). Hence $\log_a n \le O(\log_b n)$. So you don't have to write bases of logarithms in asymptotic notation, that is, you can just write $O(\log n)$. Be careful though, as this does not apply when the logarithms are in some exponent! For example, $e^{\ln(n)} = n$ but $e^{\log(n)} = n^{\log e}$, and these two expressions are certainly not of the same order.

Exercise 2.3 *O-notation quiz* **(1 point)**.

(a) For all the following functions the variable n ranges over $\mathbb{N}$. Prove or disprove the following statements. Justify your answer using Theorems 1 and/or 2.

(1) $2n^5 + 10n^2 \le O(\frac{1}{100}n^6)$

(2) $n^{10} + 2n^2 + 7 \le O(100n^9)$

(3) $e^{1.2n} \le O(e^n)$

(4)* $n^{\frac{2n+3}{n+1}} \le O(n^2)$

(b) This question does not count toward the bonus point. Find f and g as in Theorem 1 such that $f \leq O(g)$, but the limit $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ does not exist. This proves that the first point of Theorem 1 provides a sufficient, but not a necessary condition for $f \leq O(g)$.

Exercise 2.4 *Asymptotic growth of* $\sum_{i=1}^{n} \frac{1}{i}$ **(1 point)**.

The goal of this exercise is to show that the sum $\sum_{i=1}^{n} \frac{1}{i}$ behaves, up to constant factors, as $\log(n)$ when n is large. Formally, we will show $\sum_{i=1}^{n} \frac{1}{i} \leq O(\log n)$ and $\log n \leq O(\sum_{i=1}^{n} \frac{1}{i})$ as functions from $\mathbb{N}_{\geq 2}$ to $\mathbb{R}^+$.

For parts (a) to (c) we assume that $n = 2^k$ is a power of 2 for $k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We will generalise the result to arbitrary $n \in \mathbb{N}$ in part (d). For $j \in \mathbb{N}$, define

$$S_j = \sum_{i=2^{j-1}+1}^{2^j} \frac{1}{i}.$$

(a) For any $j \in \mathbb{N}$, prove that $S_j \leq 1$.

Hint: *Find a common upper bound for all terms in the sum and count the number of terms.*

(b) For any $j \in \mathbb{N}$, prove that $S_j \geq \frac{1}{2}$.

(c) For any $k \in \mathbb{N}_0$, prove the following two inequalities

$$\sum_{i=1}^{2^k} \frac{1}{i} \leq k+1$$

and

$$\sum_{i=1}^{2^k} \frac{1}{i} \geq \frac{k+1}{2}.$$

Hint: *You can use that* $\sum_{i=1}^{2^k} \frac{1}{i} = 1 + \sum_{j=1}^{k} S_j$. *Use this, together with parts (a) and (b), to prove the required inequalities.*

(d)* For arbitrary $n \in \mathbb{N}$, prove that

$$\sum_{i=1}^{n} \frac{1}{i} \leq \log_2(n) + 2$$

and

$$\sum_{i=1}^{n} \frac{1}{i} \geq \frac{\log_2 n}{2}.$$

Hint: *Use the result from part (c) for* $k_1 = \lceil \log_2 n \rceil$ *and* $k_2 = \lfloor \log_2 n \rfloor$. *Here, for any* $x \in \mathbb{R}$, $\lceil x \rceil$ *is the smallest integer that is at least* x *and* $\lfloor x \rfloor$ *is the largest integer that is at most* x. *For example,* $\lceil 1.5 \rceil = 2$, $\lfloor 1.5 \rfloor = 1$ *and* $\lceil 3 \rceil = \lfloor 3 \rfloor = 3$. *In particular, for any* $x \in \mathbb{R}$, $x \leq \lceil x \rceil < x+1$ *and* $x \geq \lfloor x \rfloor > x-1$.

Exercise 2.5 *Asymptotic growth of* $\ln(n!)$.

Recall that the factorial of a positive integer n is defined as $n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$. For the following functions n ranges over $\mathbb{N}_{\geq 2}$.

(a) Show that $\ln(n!) \leq O(n \ln n)$.

Hint: *You can use the fact that* $n! \leq n^n$ *for* $n \in \mathbb{N}_{\geq 2}$ *without proof.*

(b) Show that $n \ln n \leq O(\ln(n!))$.

Hint: *You can use the fact that* $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n!$ *for* $n \in \mathbb{N}_{\geq 2}$ *without proof.*