# Data Structures & Algorithms – HW1

Due: 14.11.2025

Honor code:

1. Do not copy the answers from any source.
2. You may work in small groups but take no written notes, and write your solution on your own (mention collaborators in the submission).

Submission guidelines:

- Submit your solution via Moodle as a PDF file **only**. Other formats will not be graded.
- Typed submissions will get a bonus of 5 points.
- If you choose not to type your solution, make sure the scan is easy to read. We will deduct points for hard-to-read submissions.

## Question 1 (15 points)

Prove the following claim using **induction**:

Claim: For all $n \geq 1$ the number $7^n + 3^n - 2$ is divisible by 8.

## Question 2 (15 points)

Let $P(n) = \frac{1}{3}n^5 - 10n^3 + 8n - 24$. Formally prove that $P(n) = \Theta(n^5)$.

Do not use the claim about polynomial that was shown in class.

## Question 3 (20 points)

Consider the following list of functions:

$f_1(n) = \log_2(2^n n^{10})$,

$f_2(n) = n^{\frac{2}{\log_2(n)}}$,

$f_3(n) = 3^{\log_2 n}$,

$f_4(n) = n/5$,

$f_5(n) = \log(n^{10})$,

$f_6(n) = 2^{3n}$,

$f_7(n) = 4^{2021} + 2020$

Arrange the following functions according to the asymptotic ordering (big O), from 'smallest' to 'biggest'. Also, find all pairs of functions for which $f_i = \Theta(f_j)$. Prove all of your claims formally.

## Question 4 (10 points):

Claim: Any group of $n \geq 8$ students, can be partitioned into non-overlapping groups of 4 or 5 students. We'll call it $P(n)$.

**Proof:**

*Base case:* For n=8: P(8) - we can divide into 2 groups of 4.

***Induction hypothesis:***
For any group k≤n, $P(k)$ can be divided into groups of 4 or 5 students.

***Induction step:***

For P(n+1), we'll form a group of 4 students. Then we can divide the remaining P(n-3) students into groups of 4 or 5 students based on the induction hypothesis.

This proves that for P(n+1), we can divide the students into groups of 4 or 5 students.

What is wrong with this proof?

## Question 5 (20 points)

For each statement below decide whether it's true or false. Prove your claim using only the definition of $\Theta(\cdot)$. **Do not use other properties shown in class.**

a. If $f(n) = \Theta(g(n))$, and for every $n$, $f(n), g(n) \geq 2$ then $\log_2(f(n)) = \Theta(\log_2(g(n)))$.
b. If $f(n) = \Theta(g(n))$, and for every $n$, $f(n), g(n) \geq 2$ then $2^{f(n)} = \Theta(2^{g(n)})$ .

Question 6 (20 points)

The function *findPair(L,k)*, takes as input a linked list *L* (whose elements are integers) and an integer *k*. The linked list is given as a pointer to the list. The function returns true if and only if there exist two different elements in L with values *x* and *y* such that *x+y = k*. The algorithm shouldn't change *L*, i.e. the linked list *L* is exactly the same during the entire execution of *FindPair*. Moreover, the algorithm should use only a constant number of pointers to list items as an additional memory. Assume that L holds at least two elements and does not contain duplicates.

Write a pseudo code for the function. Write how many comparisons will be performed by the algorithm (that is, how many times the algorithm will check whether two numbers are equal) in the best case and in the worst case as a function of *n*, the length of *L*.

A note about pseudo-code:

- Pseudo-code is intended for human reading rather than machine reading. Therefore, there are no strict rules for writing it.
- The lectures and textbook contain several examples of pseudo-code, which you can use as reference.
- The main rule of thumbs is to keep it simple and readable.
- The pseudo-code should be formal enough, so that it can easily be translated to any programming language.
- You can decide for yourself how to write pseudo-code, and we will accept any reasonable decision.