A&D

Exercise Session 1

Nil Ozer

Outline

- Logistics
- A&D Overview
- Exam
- How to study for A&D
- Get to know you

- Induction
- Rest of the exercise sheet 0

Logistics

- Written Exercise Sheets
  - Published every week on Monday (in the afternoon),
  - Hand-in deadline : Sunday, 23:59
  - Discussion on Monday during exercise session
  - Groups on Moodle
  - Peer Grading at the end of the session
- CodeExpert exercises published biweekly
  - 2 exercises
  - Youtube Videos (similar exercises from my year)
- Moodle quiz in the first ~5 minutes (from week 3)
- Bonus Grade

Logistics - for me !

- Maintaining Moodle forums for your questions

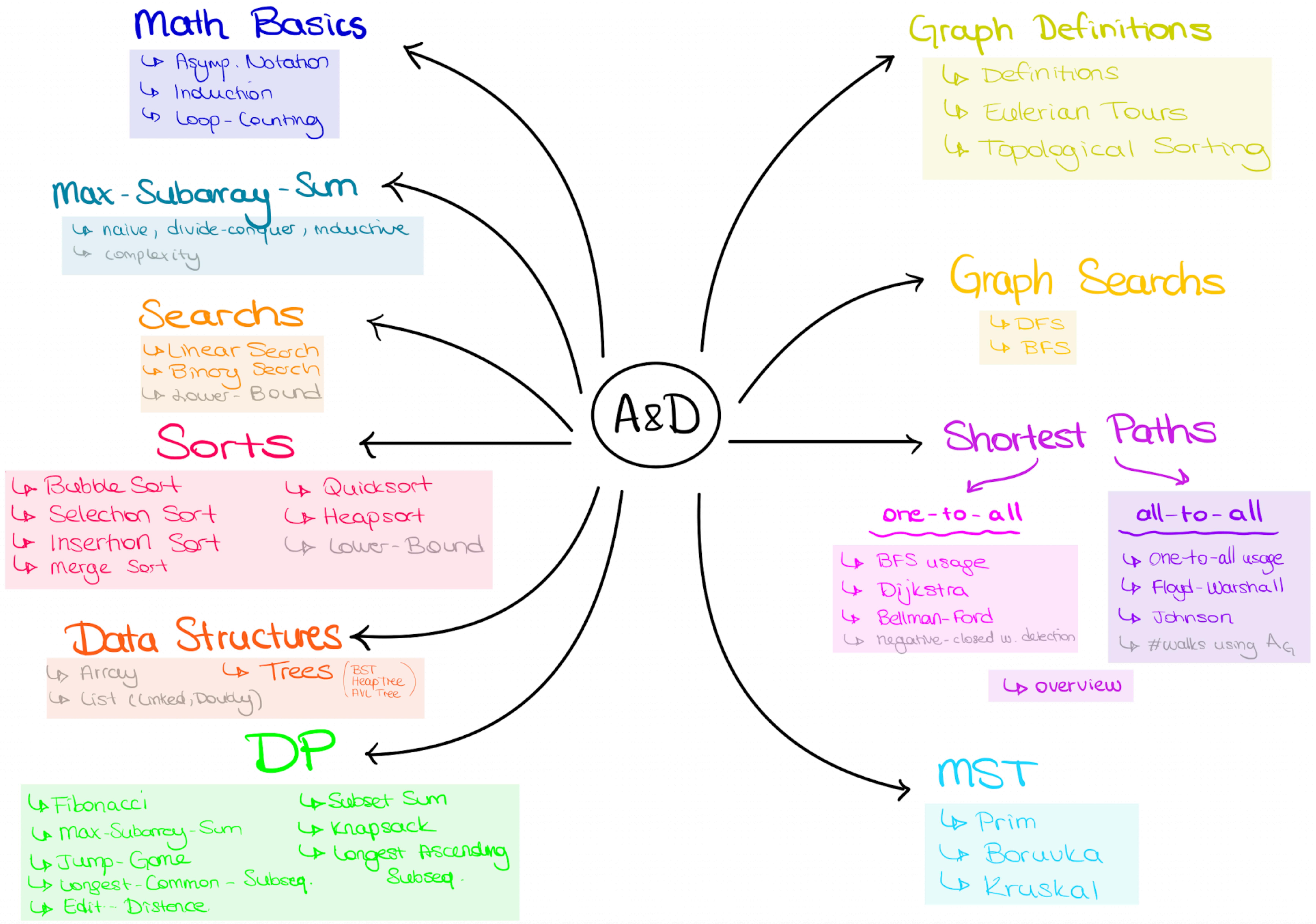
- We also have peer review
  - More consistent grading

- Weekly TA meetings with Head-TA's and Prof. Lengler
  - Discussing each exercise
  - Another way for also your voice to be heard

Website Introduction

www.nilozer.com

A&D Overview



A&D
Math Basics
Asymp. Notation
Induction
Loop-Counting
Max-Subarray-Sum
naive, divide-conquer, inductive
complexity
Searchs
Linear Search
Binary Search
Lower-Bound
Sorts
Bubble Sort
Selection Sort
Insertion Sort
Merge Sort
Quicksort
Heapsort
Lower-Bound
Data Structures
Array
List (Linked, Doubly)
Trees
BST
Heap Tree
AVL Tree
DP
Fibonacci
Max-Subarray-Sum
Jump-Game
Longest-Common-Subseq.
Edit-Distance
Subset Sum
Knapsack
Longest Ascending Subseq.
Graph Definitions
Definitions
Eulerian Tours
Topological Sorting
Graph Searchs
DFS
BFS
Shortest Paths
one-to-all
BFS usage
Dijkstra
Bellman-Ford
negative-closed w. detection
all-to-all
One-to-all usage
Floyd-Warshall
Johnson
#walks using A_G
overview
MST
Prim
Boruvka
Kruskal

Exam

Exam

Written Exam

- %60
- 2h
- VIS

Programming Exam

- %40
- 3h
- CodeExpert

	T1 (23P)	T2 (15P)	T3 (8P)	T4 (14P)	Prog. (40P)	Σ (100P)
Score						
Corrected by						

Written Exam

- T1 & T2 : Basics
- T3 : DP
- T4 : Graph

- Tasks are always similar

Programming Exam

- One DP one Graph Exercise each year
- Old exams will be published at the end of the semester

Code Expert :

Exams	DP Exercise	Graph Exercise
2022 Summer	Left and Right	Two Trees
2022 Winter	Array Compression	Tree Augmentation
2021 Summer	Pair - Subsequence	Players on a Graph
2021 Winter 1	Shortest Uncommon Subsequence	Undirected Graph
2021 Winter 2	Longest Power-of-two Subsequence	Graph Sets
2020 Summer	Longest Palindromic Subsequence	Undirected Graph
2020 Winter	Shuffle	Binary Tree

Programming Exam

Test Exam 2022 Summer

- Two Trees
- Left and Right

Left and Right

You are given an array A of n integers, indexed from 0 to $n-1$.

You play the following game. You start with a score of 0. At each step of the game, you can make one of the following moves:

1. If A contains at least two elements, you can remove the **leftmost** and the **rightmost** element of A and add to your score the absolute value of their difference. For example, if the leftmost and the rightmost elements had values x and y, you add $|x-y|$ to your score.
2. You can remove the **leftmost** element of A with no change to your score.
3. You can remove the **rightmost** element of A with no change to your score.

Your task is to find the maximum score that you can obtain in the game. You need to implement your solution as a method **getMaximumScore(**n, A**)**.

Hint: Use dynamic programming with $D[i][j]$ representing the maximum score that you can obtain on $A[i], \ldots, A[j]$.

Grading (16 points):

- An $O(n^2)$ implementation gets 16 points and an $O(n^3)$ implementation gets 6 points.

Attention: You are **NOT** allowed to use additional imports, other than the imports already included in the code template.

Two Trees

You are given two rooted trees, A and B, with disjoint vertex sets. Tree A has vertices indexed by a_0, ..., a_{n-1}, with the root at index a_0, and tree B has vertices indexed by b_0, ..., b_{n-1}, with the root at index b_0. The edges in each tree are weighted by positive integers.

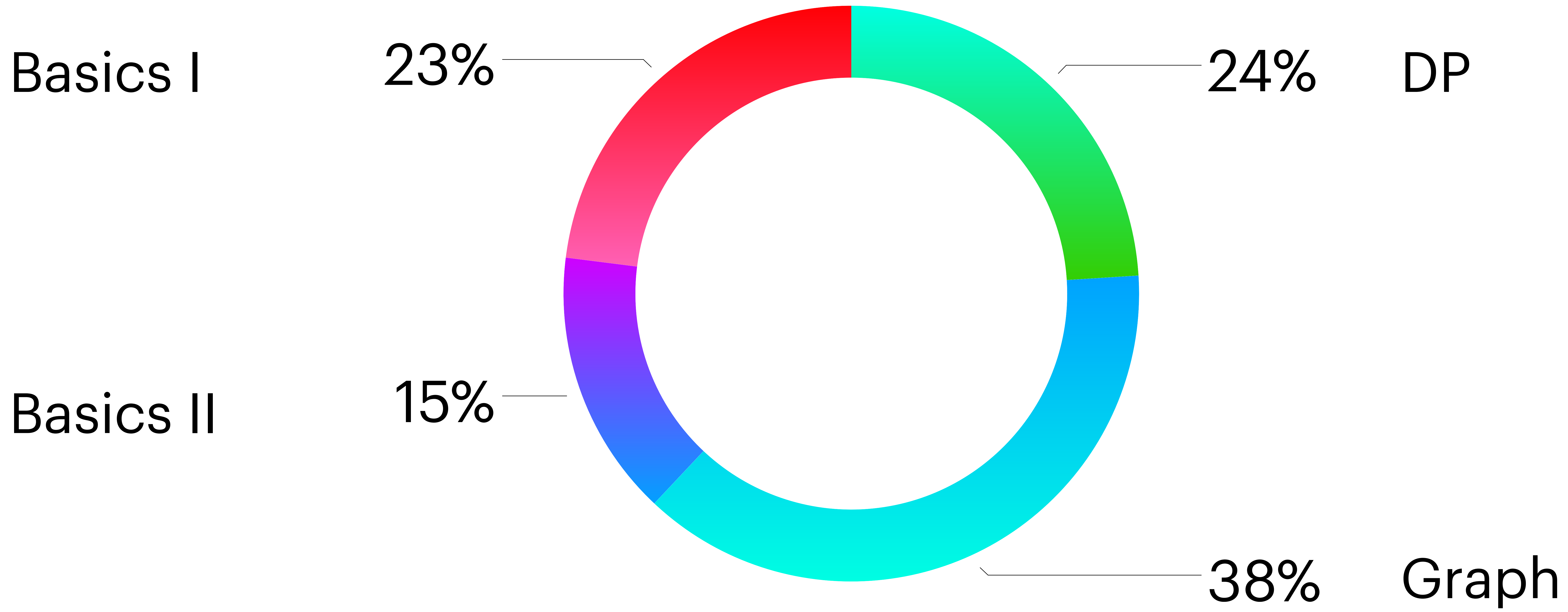
You want to add some new edges that connect leaves of A with leaves of B, thus creating a connected graph. Specifically, you can add an edge only if it goes between a leaf of A and a leaf of B. Any edge that you add has weight 0.

The distance between two vertices is defined as minimum total weight of a path that connects the two vertices.

Given these two trees, you have to implement the following methods. All the answers are guaranteed to fit on an "int" type.

1. **edgeCount()**: Return the total number of edges that you can add between the two trees.
2. **minDistRoots()**: Return the minimum distance between the roots of the two trees that you can achieve by adding exactly one edge.
3. **cycle()**: Return 1 if you can add exactly two edges such that the resulting graph has a simple cycle (no repeated vertices) that contains the two roots.
4. **minDistCycle()**: Return the minimum length of a simple cycle (no repeated vertices) that contains the two roots that you can achieve by adding exactly two edges. You can assume such a simple cycle exists.

Point Distribution



I got you !

How to study for A&D
During Semester

- Attend all lectures. If not possible, watch the lecture videos.
- Always come to the exercise session. Even if you fall back !
- Try to solve all written exercises. Work with your group !
- Ask questions ! exercise session , breaks, WhatsApp group, email , Moodle forum
- Summaries
- Feedback Feedback pools by me or contacting me directly

Get to know you

Let’s take a break

A&D 😊 👩🏻‍💻

WhatsApp group

Scan or upload this QR code using the WhatsApp camera to join this group

Induction

Summary

Importance :

* One of the most used tools in CS Area
* Appears everywhere
  - → A&D , A&W (2nd)
  - → DM
  - → FMFP (4th)
* In every written exam for A&D

Induction

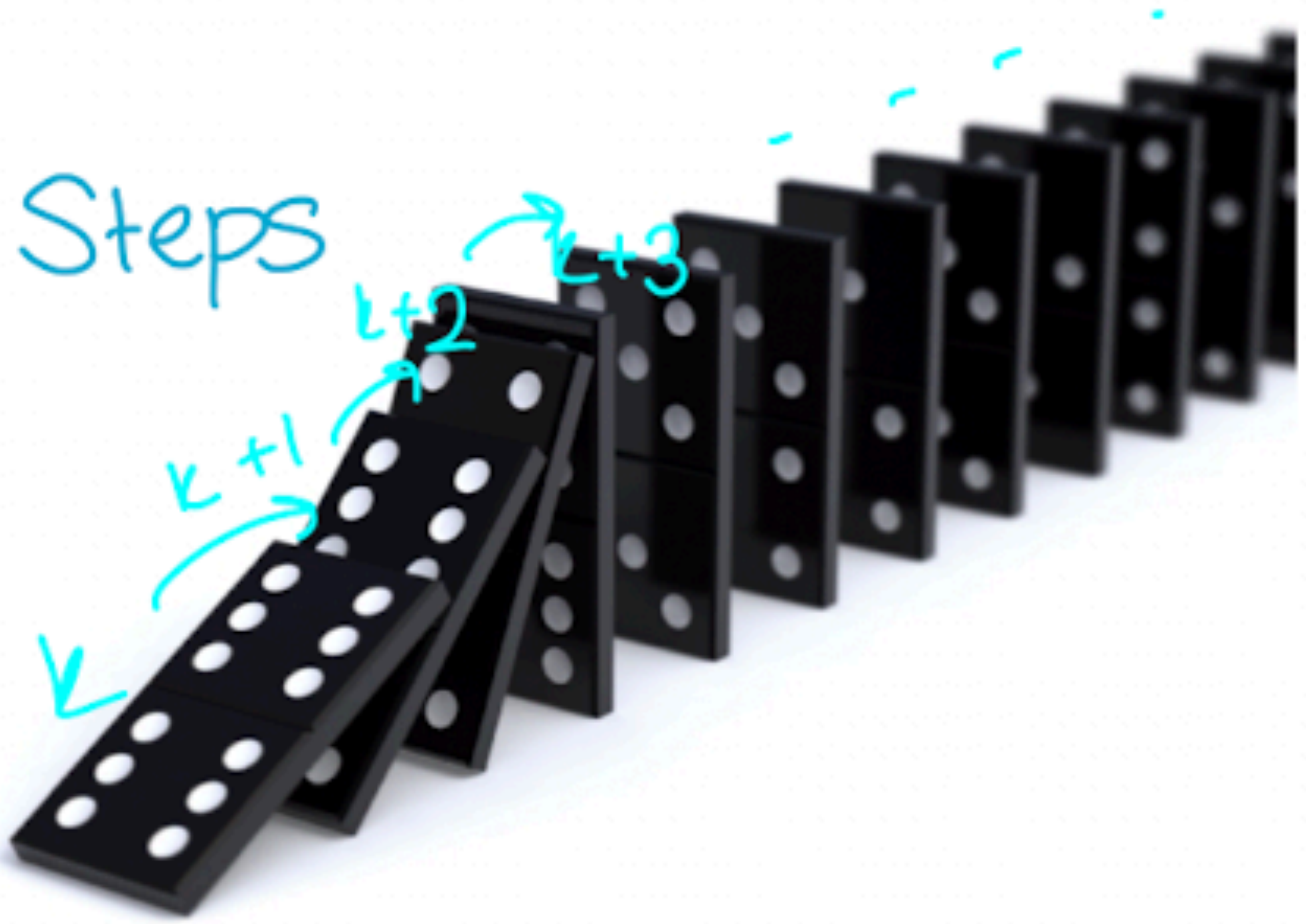
Summary

Intuition :

Domino Analogy

Base Case:

hit the first stone



Induction

Summary

Overview : $\mathcal{M}$: depends on Q

Task: Prove via induction that for every positive integer n, (for $n \geq 5$) we have

a property $\begin{cases} \dots = \dots \\ \dots \leq \dots \;/\; \dots \geq \dots \end{cases}$

(Hint : . . . .)

Base Case: Prove for $n = 1$, $(n = 5)$

Induction Hypothesis: Assume property holds for some k

Inductive Step: Show that property holds for $k+1$

Final Sentence: By the principle of mathematical induction, the property is true for every positive integer n. $(n \geq 5)$

Induction

Exercises

Exercise 0.1 *Induction.*

(a) Prove by mathematical induction that for any positive integer n,

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

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

Induction
Exercises

Exercise 0.1 *Induction.*

(a) Prove by mathematical induction that for any positive integer n,

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

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

Solution:

Base Case.
Let $n = 1$. Then we have

$$1 = \frac{1 \cdot 2}{2}.$$

Induction Hypothesis.
Assume that the property holds for some positive integer k, that is we have

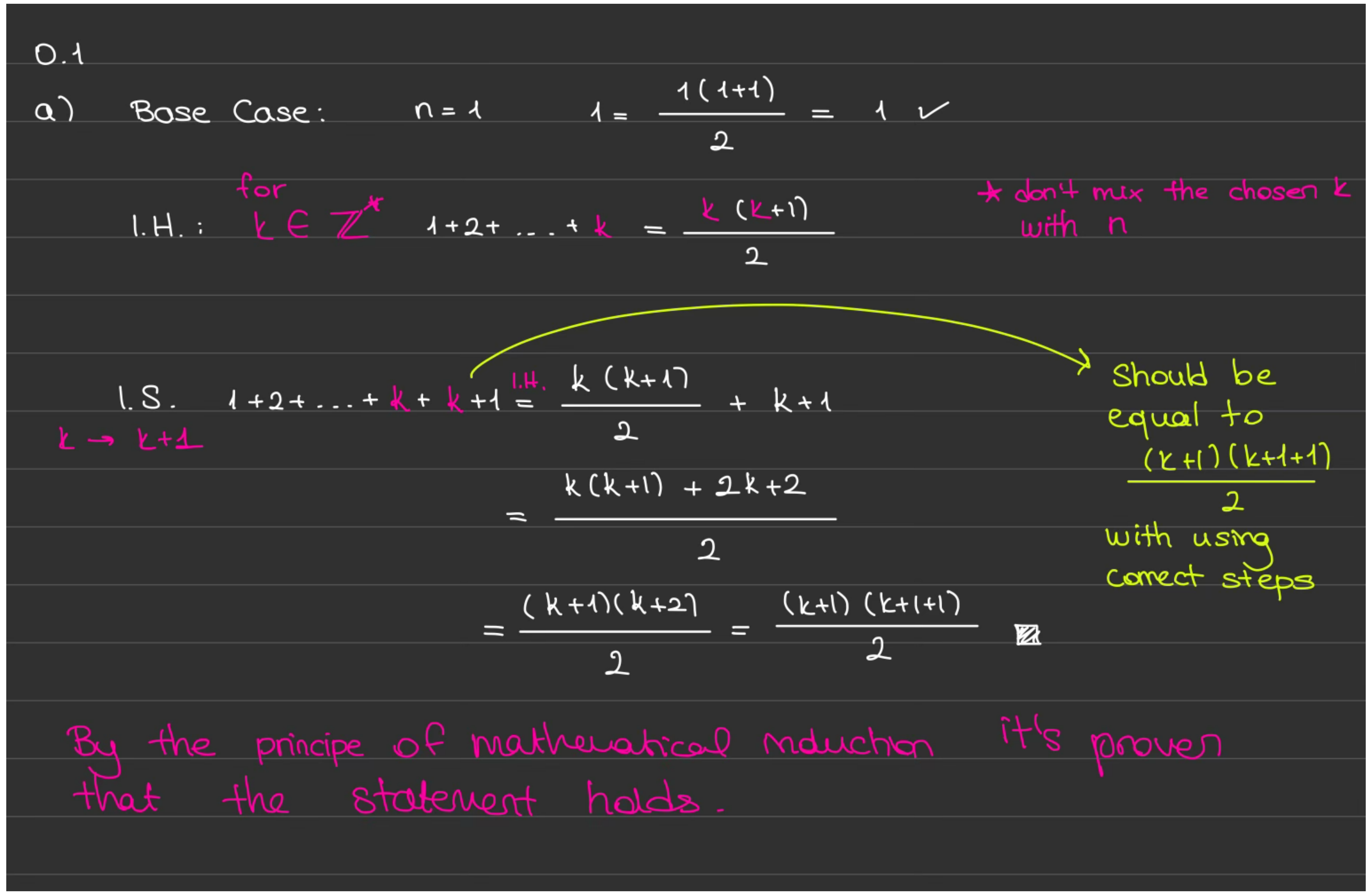
$$1 + 2 + \cdots + k = \frac{k(k+1)}{2}.$$

Induction Step.
We must show that the property holds for $k+1$ summands. We have

$$\begin{aligned} 1 + 2 + \cdots + k + (k+1) &\overset{\text{I.H.}}{=} \frac{k(k+1)}{2} + k + 1 \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2}. \end{aligned}$$

We want to use the induction hypothesis, so in the first step above we separate the last term of the sum and then use the induction hypothesis.

By the principle of mathematical induction, the statement is true for any positive integer n.



Induction

Exam question

(b) **(This subtask is from August 2019 exam).** Let $T : \mathbb{N} \to \mathbb{R}$ be a function that satisfies the following two conditions:

$$T(n) \geq 4 \cdot T(\tfrac{n}{2}) + 3n \qquad \text{whenever } n \text{ is divisible by 2;}$$
$$T(1) = 4.$$

Prove by mathematical induction that

$$T(n) \geq 6n^2 - 2n$$

holds whenever n is a power of 2, i.e., $n = 2^k$ with $k \in \mathbb{N}_0$. In your solution, you should address the base case, the induction hypothesis and the induction step.

Induction

Tipps

- Solve whenever it appears on exercise sheets
- Exam task : FS 23

/ 3 P

b) *Induction:* Consider the sequence $\{L_n\}_{n\geq 1}$ defined via $L_1 = L_2 = 1$ and the recurrence $L_{n+1} = L_n + 2L_{n-1}$ for $n \geq 2$.

Show that for all $n \in \mathbb{N} \setminus \{0\}$, the following equation holds:

$$\sum_{i=1}^{n} 2^{n-i} \cdot L_i^2 = L_n L_{n+1}.$$

Exercise Sheet 0

Rest

- Preparation for **Asymptotic-Notation quiz** tasks

/ 5 P

a) *Asymptotic notation quiz:* For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

Assume $n \geq 4$.

Claim	true	false
$n^3 + n^4 = \Theta(n^4)$	☐	☐
$n^{10} \leq O(\log(n)^{100})$	☐	☐
$1 \cdot 2 \cdot 3 \cdot \ldots \cdot n \leq O(2^n)$	☐	☐
Suppose $a_1 = 1$ and $a_{i+1} = 3a_i + 1$ for all $i \geq 2$. Then $a_n \leq O(4^n)$.	☐	☐
$2^{10\log(n)} = \Theta(2^{20\log(n)})$	☐	☐

Exercise Sheet 0
Exercise 0.2

Asymptotic Growth

When we estimate the number of elementary operations executed by algorithms, it is often useful to ignore smaller order terms, and instead focus on the asymptotic growth defined below. We denote by $\mathbb{R}^+$ the set of all (strictly) positive real numbers and by $\mathbb{R}_0^+$ the set of nonnegative real numbers.

Definition 1. Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two functions. We say that f *grows asymptotically faster than* g if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

This definition is also valid for functions defined on $\mathbb{R}^+$ instead of $\mathbb{N}$. In general, $\lim\limits_{n\to\infty} \frac{g(n)}{f(n)}$ is the same as $\lim\limits_{x\to\infty} \frac{g(x)}{f(x)}$ if the second limit exists.

For all the following exercises, you can assume that $n \in \mathbb{N}_{\geq 10}$. We make this assumption so that all functions are well-defined and take values in $\mathbb{R}^+$.

Exercise Sheet 0
Exercise 0.2

Definition 1. Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two functions. We say that f *grows asymptotically faster than* g if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

Show that

(a) $f(n) := n \log n$ grows asymptotically faster than $g(n) := n$.

Exercise Sheet 0

Exercise 0.2

Definition 1. Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two functions. We say that f *grows asymptotically faster than* g if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

Show that

(a) $f(n) := n \log n$ grows asymptotically faster than $g(n) := n$.

Solution:

We have

$$\lim_{n\to\infty} \frac{n}{n \log n} = \lim_{n\to\infty} \frac{1}{\log n} = 0$$

and hence, by Definition 1, $f(n) := n \log n$ grows asymptotically faster than $g(n) := n$.

Exercise Sheet 0
Exercise 0.3

The following theorem can be useful to compute some limits.

Theorem 1 (L'Hôpital's rule). *Assume that functions* $f : \mathbb{R}^+ \to \mathbb{R}^+$ *and* $g : \mathbb{R}^+ \to \mathbb{R}^+$ *are differentiable,* $\lim_{x\to\infty} f(x) = \lim_{x\to\infty} g(x) = \infty$ *and for all* $x \in \mathbb{R}^+$, $g'(x) \neq 0$. *If* $\lim_{x\to\infty} \frac{f'(x)}{g'(x)} = C \in \mathbb{R}_0^+$ *or* $\lim_{x\to\infty} \frac{f'(x)}{g'(x)} = \infty$, *then*

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}.$$

Exercise Sheet 0

Exercise 0.3

Definition 1. Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two functions. We say that f *grows asymptotically faster than* g if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}.$$

(b) $f(n) := e^n$ grows asymptotically faster than $g(n) := n$.

Exercise Sheet 0

Exercise 0.3

Definition 1. Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two functions. We say that f *grows asymptotically faster than* g if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}.$$

(b) $f(n) := e^n$ grows asymptotically faster than $g(n) := n$.

Solution:

We apply Theorem 1 to compute

$$\lim_{x\to\infty} \frac{x}{e^x} \overset{\text{Thm.1}}{=} \lim_{x\to\infty} \frac{x'}{(e^x)'} = \lim_{x\to\infty} \frac{1}{e^x} = 0.$$

Hence by Definition 1, $f(n) := e^n$ grows asymptotically faster than $g(n) := n$.

Exercise Sheet 0

Exercise 0.4

Exercise 0.4 *Simplifying expressions.*

Simplify the following expressions as much as possible without changing their asymptotic growth rates.

Concretely, for each expression $f(n)$ in the following list, find an expression $g(n)$ that is as simple as possible and that satisfies $\lim_{n\to\infty} \frac{f(n)}{g(n)} \in \mathbb{R}^+$.

(a) $f(n) := 5n^3 + 40n^2 + 100$

Exercise Sheet 0
Exercise 0.4

Exercise 0.4 *Simplifying expressions.*

Simplify the following expressions as much as possible without changing their asymptotic growth rates.

Concretely, for each expression $f(n)$ in the following list, find an expression $g(n)$ that is as simple as possible and that satisfies $\lim_{n\to\infty} \frac{f(n)}{g(n)} \in \mathbb{R}^+$.

(a) $f(n) := 5n^3 + 40n^2 + 100$

Let $g(n) := n^3$. Then we indeed have that

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \left(5 + \frac{40}{n} + \frac{100}{n^3}\right) = 5 \in \mathbb{R}^+.$$

Exercise Sheet 0

Rest

- 0.2 , 0.3 , 0.4
- 0.5 : * Exercise
- This was all a math background preparation

Exercise Sheet 0

Exam tipps

/ 5 P

a) *Asymptotic notation quiz:* For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

Assume $n \geq 4$.

Claim	**true**	**false**
$n^3 + n^4 = \Theta(n^4)$	☐	☐
$n^{10} \leq O(\log(n)^{100})$	☐	☐
$1 \cdot 2 \cdot 3 \cdot \ldots \cdot n \leq O(2^n)$	☐	☐
Suppose $a_1 = 1$ and $a_{i+1} = 3a_i + 1$ for all $i \geq 2$. Then $a_n \leq O(4^n)$.	☐	☐
$2^{10\log(n)} = \Theta(2^{20\log(n)})$	☐	☐

Exercise Sheet 0

Exam tipps (informal)

$\lim_{n \to \infty}$: $1 < \log(\log(n)) < \log(n) < \sqrt{n} < n < n \cdot \log(n) < n \cdot \sqrt{n} < n^2 < 2^n < n! < n^n$

$n^x < x^n$ (x being fixed)

log-Rules:

$$\begin{aligned}
\log_a(bc) &= \log_a(b) + \log_a(c) \\
\log_a(b^c) &= c\log_a(b) \\
\log_a(1/b) &= -\log_a(b) \\
\log_a(1) &= 0 \\
\log_a(a) &= 1 \\
\log_a(a^r) &= r \\
\log_{1/a}(b) &= -\log_a(b) \\
\log_a(b)\log_b(c) &= \log_a(c) \\
\log_b(a) &= \frac{1}{\log_a(b)} \\
\log_{a^m}(a^n) &= \frac{n}{m}, \qquad m \neq 0
\end{aligned}$$

$$\log_b a = \frac{\log_d a}{\log_d b}$$

Questions

Feedbacks , Recommendations

Nil Ozer