Week 1
In(tro)duction

Thorben Klabunde

22.09.2025

Agenda

Meet & Greet

About Me

- **Name:** Thorben Klabunde
- **Hometown:** Winterthur, CH
- **Hobbies:** Swimming & cooking

About You

- Now it's your turn so we all get to know each other!
- Please briefly introduce yourself:
  - **Name**
  - **Hometown**
  - **Hobbies**

About the Course

Organizational

- **Schedule & Contact**
  - **Schedule**: LEE C104, Mo. 09:00-12:00
  - **Contact**: tklabunde@student.ethz.ch
  - **Resources & materials**: www.th-kl.ch

Organizational

- **Exercises & Bonus Points**

  During the semester, you can get bonus points for

  - solving the designated parts of the **theoretical exercise sheets** (in working groups);
    3 points per exercise sheet;
  - solving the **multiple choice quizzes** in the beginning of the exercise class (individually);
    1 point per week starting in week 3;
  - **peer grading** the specified part of the theory sheets (in working groups);
    1 point per exercise sheet;
  - solving the **programming problems** (individually);
    6 points per exercise sheet.

  You can find all details regarding grading on the Moodle Course Website. Please make sure to read it carefully!

Organizational

- **Exercises & Bonus Points**

  BUT don't stress out too much!

  - You only need **80% of the points** (!) and even if you have no bonus points, you can still get the highest mark.
  - **Code-Expert** is worth a lot of points, which you have plenty of time for.
  - **Quizzes** give you valuable feedback but are **only worth 1 point** (I certainly didn't have full marks on quite a few quizzes, no need to worry!)
  - **Assignments do not need to be as formal as the master solution**. I will try to guide you and give you feedback.

Organizational

- **Feedback-System:**

  Although it might seem like a lot, the system is designed to help you!

  - **Working-Groups:**

    Partners for assignment submissions (newly assigned by me every 3 weeks).

    Solutions to exercise sheets are submitted in pairs (only one submits solution on Moodle).

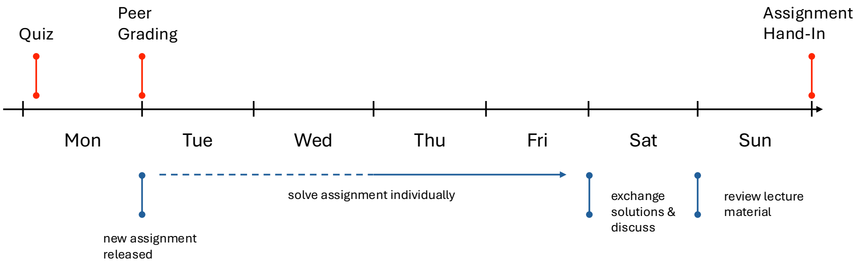
    Use the chance to get to know some of your peers!

  - **Peer-Grading:**

    Forces you to take an outside perspective and notice what assumptions to make explicit.

Weekly Schedule for A&D

This ended up being my schedule for A&D last fall just to give you a sense. At the start of the term, you'll have a lot more time and will finish assignments much sooner. Don't worry when it shifts back slightly!



A&D Overview

Weeks

1-5

Basics & Complexity

- Asympt. Notation
- Induction
- Loop Counting

Search

- Linear Search
- Binary Search
- Lower Bound

Sorting

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Lower Bound

5-7

Data Structures

ADTs

- List, Stack, Queue, Dictionary

Data Struct.

- Arrays, Lists, Heaps, Trees

Dynamic Programming

- Fibonacci Numbers
- Max. Subarray Sum
- Jump Game
- Longest Common Subseq.
- Edit Dist.
- Subset Sum
- Knapsack
- Longest Ascending Subseq.

8-14

Graph Theory

Introduction

- Definitions & Properties
- Topological Sort

Graph Search

- Depth First Search – DFS
- Breadth First Search – BFS

Shortest Path

Single-source SP

- BFS
- Djikstra
- Bellman-Ford

All-pair SP

- Floyd-Warshall
- Johnson
- Matrix Mult.

Minimum Spanning Trees

- Prim
- Boruvka
- Kruskal

Exam

- **4h split btw. theory and coding** - the exam format has been revised this term, I will update you once I know more
- The **assignments ideally prepare you for the exam**! Try to do all of them!

Questions?

Website: www.th-kl.ch **Feedback:** Google Forms (anonymous)

Theory Recap

Karatsuba's Algorithm: Idea & Correctness

- **Goal:** Multiply two n-digit numbers faster than the classical, grade-school approach, i.e., using **fewer single-digit multiplications**.

Karatsuba's Algorithm: Idea & Correctness

- **Goal:** Multiply two n-digit numbers faster than the classical, grade-school approach, i.e., using **fewer single-digit multiplications**.
- **Derivation:**

  Let x and y be two n digit numbers, where $n > 1$ is a power of 2 (simplifying assumption), and let $m = n/2$.

- **Derivation:**

  Let x and y be two n digit numbers, where $n > 1$ is a power of 2 (simplifying assumption), and let $m = n/2$.

  1. Notice that we can deconstruct x and y into:

$$x = a \cdot 10^m + b \quad \text{and} \quad y = c \cdot 10^m + d$$

Karatsuba's Algorithm: Idea & Correctness

- **Derivation:**

  Let x and y be two n digit numbers, where $n > 1$ is a power of 2 (simplifying assumption), and let $m = n/2$.

  1. Notice that we can deconstruct x and y into:

  $$x = a \cdot 10^m + b \quad \text{and} \quad y = c \cdot 10^m + d$$

  2. Factoring out yields:

  $$xy = ac \cdot 10^{2m} + (ad + bc) \cdot 10^m + bd$$

Karatsuba's Algorithm: Idea & Correctness

- **Derivation:**

  Let x and y be two n digit numbers, where $n > 1$ is a power of 2 (simplifying assumption), and let $m = n/2$.

  1. Notice that we can deconstruct x and y into:

  $$x = a \cdot 10^m + b \quad \text{and} \quad y = c \cdot 10^m + d$$

  2. Factoring out yields:

  $$xy = \mathbf{ac} \cdot 10^{2m} + (\mathbf{ad} + \mathbf{bc}) \cdot 10^m + \mathbf{bd}$$

  But this still needs **4 multiplications**

Karatsuba's Algorithm: Idea & Correctness

- **Key Idea:** Notice that

$$
\begin{aligned}
(a+b)(c+d) &= ac + ad + bc + bd && (1)\\
&= ac + (ad + bc) + bd && (2)\\
\overset{\text{arithm. rearr.}}{\iff} \quad (\mathbf{ad} + \mathbf{bc}) &= (a+b)(c+d) - \mathbf{ac} - \mathbf{bd} && (3)
\end{aligned}
$$

Only **3 multiplications** required!

Pasture Break

Setting: A shortsighted cow is at an 'origin' point on a very long, circular fence of length l.

Goal: Find the closest gap in the fence.

Constraints: The closest gap is k steps away from the origin, but k is unknown. The total fence length l is much larger than k ($l \gg k$). The cow can only see the gap when standing directly next to it.

Strategy I: Linear Step-Sizing

- Proceed in a zig-zag pattern, increasing the length one step at a time.
- **Number of Steps (worst case):**

$$2\cdot(1+2+3+\dots+k)+k$$

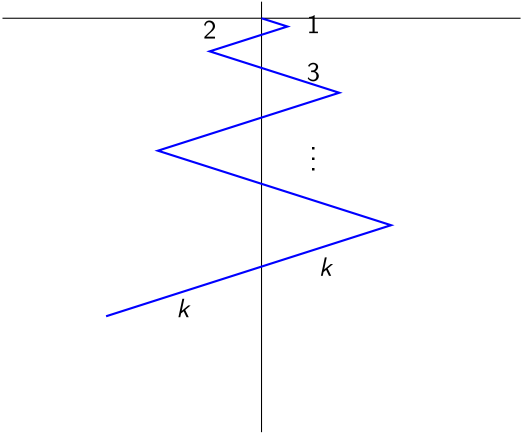
$$=2\cdot\left(\sum_{i=1}^{k} i\right)+k$$

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

$$=k(k+1)+k$$

$$=k(k+1+1)$$

$$=k(k+2)=k^2+2k$$

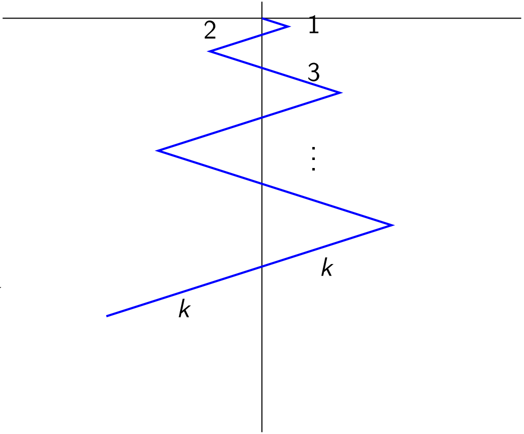


Pasture Break: Linear Step-Sizing

Strategy I: Linear Step-Sizing

- Proceed in a zig-zag pattern, increasing the length one step at a time.
- **Number of Steps (worst case):**

$$
\begin{aligned}
2 * (1 + 2 + \ldots + (k-1) + k) + k &= 2 \cdot \left(\sum_{i=1}^{k} i\right) + k \\
&= 2 \cdot \frac{k \cdot (k+1)}{2} + k \\
&= k^2 + 2k
\end{aligned}
$$

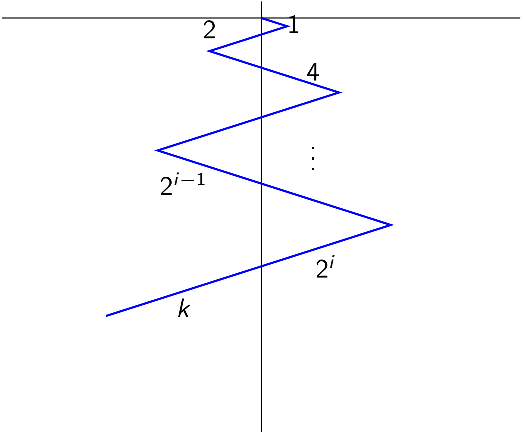


Pasture Break: Exponential Step-Sizing

Strategy II: Exponential Increase in Step-Size

- Proceed in a zig-zag pattern, doubling the search distance after a failed attempt.
- **Number of Steps (worst case):**

$$2\cdot(2^0+2^1+2^2+\dots+2^i)+k$$
$$= 2\cdot(2^{i+1}-1)+k$$
$$\le 2\cdot(2^2\cdot 2^{(i-1)})+k$$
$$\le 2^3\cdot k + k$$
$$= 9k$$



Strategy II: Exponential Increase in Step-Size

- Proceed in a zig-zag pattern, doubling the search distance after a failed attempt.
- **Number of Steps (worst case):**

$$2 \cdot (1 + 2 + \ldots + 2^i) + k = 2 \cdot \left(\sum_{l=0}^{i} 2^l\right) + k$$

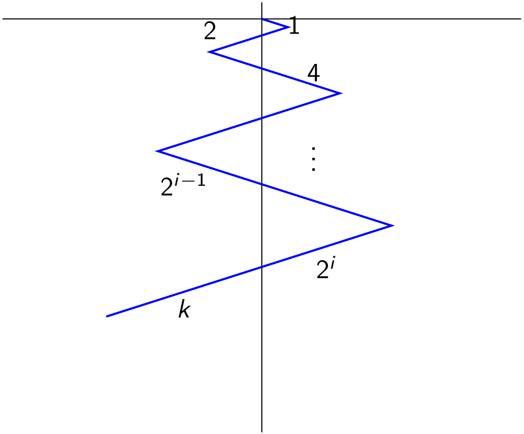
$$\overset{\text{Geo. Sum}}{=} 2 \cdot (2^{i+1} - 1) + k$$

$$\leq 2 \cdot (2^2 \cdot 2^{i-1}) + k$$

Using that $2^{i-1} \leq k$:

$$\leq 8 \cdot k + k$$

$$= 9k$$



Recall: Finite Geo. Sum

$\sum_{l=0}^{n} r^l = \frac{r^{n+1}-1}{r-1}, \quad r \neq 1$

Proof by Induction - baby steps go a long way!

Recall: When do we use a proof by induction? How do we proceed?

Proof by Induction - baby steps go a long way!

Proof By Induction

Used to prove statements of the form $\forall n \; P(n)$, where the universe is the set $\{k, k+1, k+2, \ldots\}$ for some $k \in \mathbb{N}$.

Consists of **two steps**:

Proof by Induction - baby steps go a long way!

Proof By Induction

Used to prove statements of the form $\forall n \; P(n)$, where the universe is the set $\{k, k+1, k+2, \ldots\}$ for some $k \in \mathbb{N}$.

Consists of **two steps**:

1. **Basis step:** Prove $P(k)$.
2. **Induction step:** Prove that for arbitrary $n \geq k$, $P(n) \implies P(n+1)$.

The induction step is performed by assuming $P(n)$ (the **induction hypothesis**) and deriving $P(n+1)$.

Proof by Induction - A Strategic Workflow

1. **Formulate the claim**: What are we trying to prove?
2. **Identify the Induction Variable**: Determine the variable, n, and its starting value, k.
3. **Identify the Base Case(s) (B.C.)**

   Think ahead! The structure of the Induction Step (I.S.) determines how many base cases you must prove.
4. **State the Induction Hypothesis (I.H.)**
   - *Weak Induction*: Assume $P(n)$ is true for an arbitrary integer $n \geq k$.
   - *Strong Induction*: Assume $P(i)$ is true for all integers i such that $k \leq i \leq n$.
5. **Prove the Base Case(s)**

   Beware, sometimes there are multiple.
6. **Prove the I.S.** under the assumption of the I.H.

Assignment

Exercise 0.1.a)

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.

cl. For arb. $n \in \mathbb{N}^+$ it holds: $1+2+\ldots+n = \frac{n(n+1)}{2}$

pr. We proceed by induction on n.

B.C. Let $n=1$. We have: $1 = \frac{1(1+1)}{2}$ and the B.C. holds.

I.H. Assume that the claim holds for some $n \geq 1$.

Exercise 0.1.a) - continued

I.S. Then for $n+1$ we have:

$$1+2+\dots+n+(n+1) = \underbrace{(1+2+\dots+n)}_{\text{I.H.}} + (n+1)$$

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

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

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

By the princ. of math. induction, the claim holds for all $n \in \mathbb{N}^+$. ∎

Exercise 0.1.b)

(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.

Exercise 0.1.b) - continued

Let $T: \mathbb{N} \to \mathbb{R}$ satisfy: $T(n) \geq 4T(\frac{n}{2}) + 3n$, $T(1) = 4$

Cl. For $n = 2^k$ with $k \in \mathbb{N}_0$ it holds: $T(n) \geq 6n^2 - 2n$.

Pr. We proceed by induction on k.

B.C. Let $k = 0$, wherefore $n = 2^0 = 1$. It holds $T(1) = 4 \geq 6 \cdot 1 - 2 \cdot 1$. ✓

I.H. Assume for some $k \in \mathbb{N}^0$ the claim holds.

I.S. Then for $k+1$ we have:

$$T(2^{k+1}) \geq 4 \cdot T(2^k) + 3 \cdot (2^{k+1}) \quad \text{| (by assumption)}$$

$$\overset{\text{I.H.}}{\geq} 4 \cdot (6 \cdot (2^k)^2 - 2 \cdot 2^k) + 3 \cdot (2^{k+1})$$

$$= 6 \cdot 4 \cdot (2^k)^2 - 8 \cdot 2^k + 3 \cdot (2^{k+1})$$

$$= 6 \cdot 2^2 \cdot (2^k)^2 - 4 \cdot (2^{k+1}) + 3 \cdot (2^{k+1})$$

$$= 6 \cdot (2^{k+1})^2 - 1 \cdot (2^{k+1})$$

$$\geq 6 \cdot (2^{k+1})^2 - 2 \cdot 2^{k+1} \quad \text{| (since } 2^{k+1} \geq 0 \text{ for all } k \in \mathbb{N}_0)$$

Take a step back & remember where we went to go.

Tip: Sometimes it helps to jot down the final step & work it backwards.

By the princ. of math. induction the claim holds for all $k \in \mathbb{N}_0$. ∎

Exercise 0.2.b)

(b) $f(n) := n^3$ grows asymptotically faster than $g(n) := 10n^2 + 100n + 1000$.

To show: $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$ ($\Leftrightarrow$ "$f(n)$ grows asymptotically faster than $g(n)$")

(b) $\frac{g(n)}{f(n)} = \frac{10n^2 + 100n + 1000}{n^3} = \frac{n^3}{n^3} \cdot \frac{\frac{10}{n} + \frac{100}{n^2} + \frac{1000}{n^3}}{1}$

$\Rightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{10}{n} + \frac{100}{n^2} + \frac{1000}{n^3} \overset{\text{(all limits exist)}}{=} \lim_{n\to\infty} \frac{10}{n} + \lim_{n\to\infty} \frac{100}{n^2} + \lim_{n\to\infty} \frac{1000}{n^3} = 0$

Exercise 0.2.c)

(c) $f(n) := 3^n$ grows asymptotically faster than $g(n) := 2^n$.

(c) $$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{2^n}{3^n} \overset{(\text{exp. laws})}{=} \lim_{n\to\infty} \underbrace{\left(\frac{2}{3}\right)^n}_{-1 < \frac{2}{3} < 1} = 0$$

Exercise 0.3.a)

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*

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

Hôpital can come in handy when computing limits.

However, make sure to check that the conditions of the theorem are satisfied before applying it (make a brief remark, expl. why they hold).

Exercise 0.3.a)

given assumption: $n \in \mathbb{N}_{n \geq 10}$

(a) $f(n) := n^{1.01}$ grows asymptotically faster than $g(n) := n \ln n$.

(a) $$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{n \ln n}{n^{1.01}} = \lim_{n\to\infty} \frac{\ln n}{n^{0.01}} \overset{\text{Hôp.}^*}{=} \lim_{n\to\infty} \frac{\frac{1}{n}}{0.01 \cdot n^{-0.99}}$$

$$= \lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{0.01}{n^{0.99}}} = \lim_{n\to\infty} \frac{n^{0.99}}{0.01 \cdot n} = \lim_{n\to\infty} \frac{1}{0.01 \cdot n^{0.01}} = 0$$

* Notice that: (1) $\ln(n)$ & $n^{0.01}$ are diff. for $n \geq 10$, (2) $\lim_{n\to\infty} \ln(n) = \lim_{n\to\infty} n^{0.01} = \infty$

& (3) $\forall n \geq 10 \; (0.01 \cdot n^{-0.99} \neq 0) \Rightarrow$ Hôpital is applicable.

(see slide before)

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

(b) $\lim_{n \to \infty} \frac{n}{e^n} \overset{*\,\text{Hôp}}{=} \lim_{n \to \infty} \frac{1}{e^n} = 0$

* Notice that Hôpital is applicable since (i) n & e^n are diff. on $\mathbb{R}$, (ii) $\lim_{n \to \infty} n = \lim_{n \to \infty} e^n = \infty$ and (iii) $\forall n \in \mathbb{N} \left(g'(n) \neq 0 \right)$.

Exercise 0.3.d)

(d)* $f(n) := 1.01^n$ grows asymptotically faster than $g(n) := n^{100}$.

$$\lim \frac{g(n)}{f(n)} = \lim \frac{n^{100}}{1.01^n} = \lim e^{\ln\left(\frac{n^{100}}{1.01^n}\right)} \overset{\left(\substack{\text{exp.}\\ \text{is cont.}}\right)}{=} e^{\lim \ln\left(\frac{n^{100}}{1.01^n}\right)}$$

Recall, $\forall x \in \mathbb{R}_+ \forall r \in \mathbb{R}$:
$\log(x^r) = r \cdot \log(x)$

Consider only the exponent: $\ln \frac{n^{100}}{1.01^n} = \ln(n^{100}) - \ln(1.01^n)$
$= 100 \cdot \ln(n) - n \cdot \ln(1.01)$

Notice: $\lim\limits_{n\to\infty} 100 \cdot \ln(n) - n \cdot \ln(1.01) = \lim\limits_{n\to\infty} n \cdot \Big(\underbrace{\frac{100 \cdot \ln(n)}{n}}_{\to 0} - \ln(1.01) \Big) = -\infty$

It follows that $\lim\limits_{n\to\infty} e^{\ln\left(\frac{n^{100}}{1.01^n}\right)} = 0$. ∎

Exercise 0.3.f)

(f) $f(n) := 2^{\sqrt{\log_2 n}}$ grows asymptotically faster than $g(n) := \log_2^{100} n$.

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{(\log_2 n)^{100}}{2^{\sqrt{\log_2 n}}} = \lim_{n\to\infty} \frac{2^{\log_2(\log_2^{100} n)}}{2^{\sqrt{\log_2 n}}} \overset{(IV)}{=} \lim_{n\to\infty} 2^{\log_2(\log_2^{100} n) - \sqrt{\log_2 n}} \overset{(2^x \text{ is cont. on } \mathbb{R})}{=} 2^{\lim_{n\to\infty}(\dots)}$$

exponentials are quite nice to work with
$\Rightarrow$ get both terms to a common base & focus on exponent

(I) $\log\left(\frac{a}{b}\right) = \log(a) - \log(b)$
(II) $\log(a \cdot b) = \log(a) + \log(b)$
(III) $2^a \cdot 2^b = 2^{a+b}$
(IV) $\frac{2^a}{2^b} = 2^{a-b}$

Consider the exp. and notice:

$$\lim_{n\to\infty} \log_2(\log_2^{100}(n)) - \sqrt{\log_2 n} = \lim_{n\to\infty} 100 \cdot \log_2 \underbrace{\log_2(n)}_{x} - \sqrt{\underbrace{(\log_2 n)}_{x}}$$

$$= \lim_{x\to\infty} 100 \cdot \log_2(x) - \sqrt{x} = \lim_{x\to\infty} \underbrace{\sqrt{x}}_{\to\infty} \underbrace{\left(\frac{100 \cdot \log_2(x)}{\sqrt{x}} - 1\right)}_{\to -1} = -\infty \Rightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$

Exercise 0.4

We did not get to discuss ex. 0.4 in detail in class.

However, the master solutions are very detailed, please refer to them!

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 $\lim_{n \to \infty} \frac{f(n)}{g(n)} \in \mathbb{R}^+$.

Exercise 0.4.d)

(d) $f(n) := 23n + 4n \log_5 n^6 + 78\sqrt{n} - 9$

Please refer to master solution.

Exercise 0.4.e)

(e) $f(n) := \log_2 \sqrt{n^5} + \sqrt{\log_2 n^5}$

Please refer to master solution.

Exercise 0.4.f)

(f)* $f(n) := 2n^3 + (\sqrt[4]{n})^{\log_5 \log_6 n} + (\sqrt[7]{n})^{\log_8 \log_9 n}$

Please refer to master solution.

Tips & Tricks

Workflow

Disclaimer: This is my workflow and is based on my experience and what works for me. I hope it helps you discover a suitable workflow for yourself. Don't let anyone's opinions sway you too much if you've found something that works.

Note-Taking Style

- Active-Recall using Question-Answer style notes is great for revision
- Would recommend a structured note-taking app (see below) and advise against flash-card apps like Anki

  Used Anki in first term and found that it lacked structure for look-ups and revision

Note-Taking App:

Notion lets you do both - structured note-taking & active-recall style notes (with toggled bullets)

- **Pros:** Easy, clean, well-formatted note-taking with nice shortcuts and features (inline latex, code-blocks with syntax highlighting, toggle bullets, search)
- **Cost:** **Free** - Sign-up with your student mail to get a free Plus Plan.

Workflow - Handwritten Notes

GoodNotes

- **Pros:** Clean and well-structured user-interface, nice pens & templates.
- **Cost: 10CHF/year (iOS)** - Not free but reasonable pricing.