ETH zürich

Exercise Session 01 – Asymptotics

Data Structures and Algorithms

These slides are based on those of the lecture, but were adapted and extended by the teaching assistant Adel Gavranović

Today's Schedule

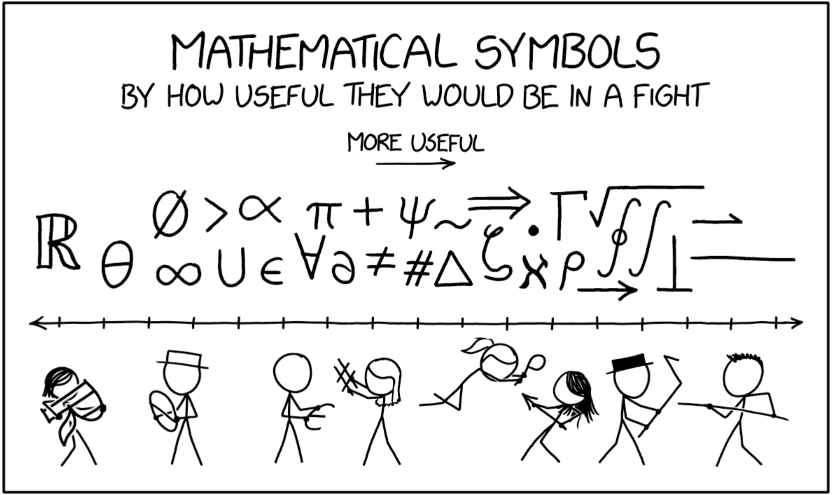
Intro
Learning Objectives
Exercise Process
Repetition Theory
- Examples (Theory)
- Asymptotic Running Time of Program Fragments

Tips for **code** expert
Outro

`n.ethz.ch/~agavranovic`

- Exercise Session Material
- Adel's Webpage
- Mail to Adel

Comic of the Week



▸ xkcd

1. Intro

Intro

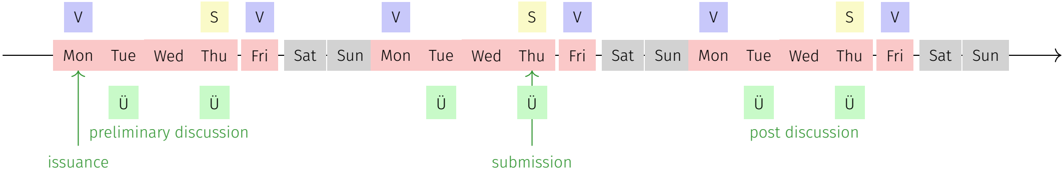
- Who am I?
- Who are you?

2. Learning Objectives

Learning Objectives

- ☐ Get to know the weekly agenda for this course
- ☐ Understand differences between Problem, Algorithm, and Program
- ☐ Get to know big-$\mathcal{O}$ notation (and its friends Ω and Θ)
- ☐ Learn some LaTeX and Markdown

Exercises



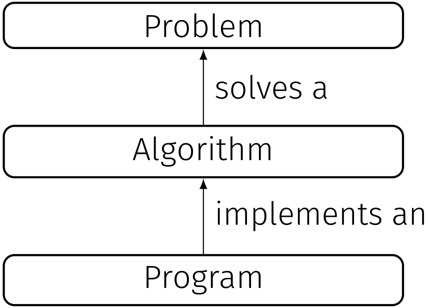
- Exercises availabe on Monday.
- Preliminary discussion in the following recitation session
- Solution of the exercise until the following Thursday.
- Discussion of the exercise in the next recitation session.
- Feedback roughly within 10 days after submission date.
- Study Center on Thursday.

4. Repetition Theory

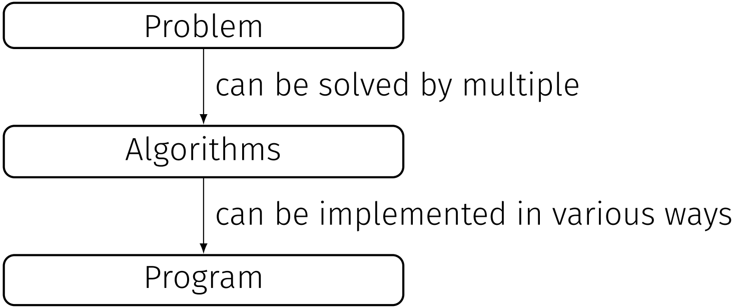
Warm-up

- What is a problem?
- What is an algorithm?
  - ➜ well-defined computing procedure to compute output data from input data.
- What is a program?
  - ➜ Concrete implementation of an algorithm

Problems, Algorithms and Programs



Warm-up


Problem
can be solved by multiple
Algorithms
can be implemented in various ways
Program

Efficiency

Program	Computing time	Measurable value on an actual machine.
Algorithm	Cost	Number of elementary operations
Problem	Complexity	Minimal (asymptotic) cost over all algorithms that solve the problem.

➜ Estimation of cost or computing time depending on the input size, denoted by n.

Asymptotic behavior

- What are $\Omega(g(n))$, $\Theta(g(n))$, $\mathcal{O}(g(n))$?

→ Sets of functions!

subset	$A \subseteq B$
proper subset	$A \subsetneq B$
intersection	$A \cap B$

Asymptotic behavior

Given: function $f : \mathbb{N} \to \mathbb{R}$.

Definition:

$$\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$
$$\Omega(g) = \{f : \mathbb{N} \to \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq c \cdot g(n) \leq f(n)\}$$
$$\Theta(g) = \mathcal{O}(g) \cap \Omega(g)$$

Intuition:
$f \in \mathcal{O}(g)$: f grows asymptotically not faster than g. Algorithm with running time f is not worse than any other algorithm with g.
$f \in \Omega(g)$: f grows asymptotically not slower than g. Algorithm with running time f is worse than any other algorithm with g.
$f \in \Theta(g)$: f grows asymptotically as fast as g. Algorithm with running time f is as good as any other algorithm with g.

Used less often

Given: function $f : \mathbb{N} \rightarrow \mathbb{R}$.
Definition:

$$\mathcal{O}(g) = \{f : \mathbb{N} \rightarrow \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$

$$o(g) = \{f : \mathbb{N} \rightarrow \mathbb{R} | \forall c > 0 \ \exists n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$

$$\Omega(g) = \{f : \mathbb{N} \rightarrow \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq c \cdot g(n) \leq f(n)\}$$

$$\omega(g) = \{f : \mathbb{N} \rightarrow \mathbb{R} | \forall c > 0 \ \exists n_0 \in \mathbb{N} | \forall n \geq n_0 : 0 \leq c \cdot g(n) \leq f(n)\}$$

$f \in o(g)$: f grows much slower than g
$f \in \omega(g)$: f grows much faster than g

Useful information for the exercise

Theorem 1

1. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f \in \mathcal{O}(g)$, $\mathcal{O}(f) \subsetneq \mathcal{O}(g)$.
2. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C > 0$ (C *constant*) $\Rightarrow f \in \Theta(g)$.
3. $\frac{f(n)}{g(n)} \underset{n\to\infty}{\longrightarrow} \infty \Rightarrow g \in \mathcal{O}(f)$, $\mathcal{O}(g) \subsetneq \mathcal{O}(f)$.

Example 2

1. $\lim_{n\to\infty} \frac{n}{n^2} = 0 \Rightarrow n \in \mathcal{O}(n^2)$, $\mathcal{O}(n) \subsetneq \mathcal{O}(n^2)$.
2. $\lim_{n\to\infty} \frac{2n}{n} = 2 > 0 \Rightarrow 2n \in \Theta(n)$.
3. $\frac{n^2}{n} \underset{n\to\infty}{\longrightarrow} \infty \Rightarrow n \in \mathcal{O}(n^2)$, $\mathcal{O}(n) \subsetneq \mathcal{O}(n^2)$.

Property

$$f_1 \in \mathcal{O}(g), f_2 \in \mathcal{O}(g) \Rightarrow f_1 + f_2 \in \mathcal{O}(g)$$

4.1 Examples (Theory)

Examples

$$\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R} | \; \exists c > 0, \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)\}$$

$f(n)$	$f \in \mathcal{O}(?)$	Example
$3n + 4$	$\mathcal{O}(n)$	$c = 4, n_0 = 4$
$2n$	$\mathcal{O}(n)$	$c = 2, n_0 = 0$
$n^2 + 100n$	$\mathcal{O}(n^2)$	$c = 2, n_0 = 100$
$n + \sqrt{n}$	$\mathcal{O}(n)$	$c = 2, n_0 = 1$

Examples

- $n \in \mathcal{O}(n^2)$ correct, but too imprecise:
  $n \in \mathcal{O}(n)$ and even $n \in \Theta(n)$.
- $3n^2 \in \mathcal{O}(2n^2)$ correct but uncommon:
  Omit constants: $3n^2 \in \mathcal{O}(n^2)$.
- $2n^2 \in \mathcal{O}(n)$ is wrong: $\frac{2n^2}{n} = 2n \underset{n\to\infty}{\to} \infty$!
- $\mathcal{O}(n) \subseteq \mathcal{O}(n^2)$ is correct
- $\Theta(n) \subseteq \Theta(n^2)$ is wrong $n \notin \Omega(n^2) \supset \Theta(n^2)$

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$?	✓
$2^n \notin \mathcal{O}(\exp(n))$?	✗

A good strategy?

... Then I simply buy a new machine! If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ... [1]

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$	$n \to n^{10}$	$n \to n^{100}$
n	$n \to 10 \cdot n$	$n \to 100 \cdot n$
n^2	$n \to 3.16 \cdot n$	$n \to 10 \cdot n$
2^n	$n \to n + 3.32$	$n \to n + 6.64$

[1] To see this, you set $f(n') = c \cdot f(n)$ ($c = 10$ or $c = 100$) and solve for n'

4.2 Asymptotic Running Time of Program Fragments

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i)
  op();
}
```

How often is `op()` called as a function of n?

$$\sum_{i=1}^{n-1} 1 = n - 1 \in \Theta(n)$$

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i)
    for (int j = 1; j<n; ++j)
      op();
}
```

How often is `op()` called as a function of n?

$$\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} 1 = \sum_{i=1}^{n-1}(n-1) = (n-1)\cdot(n-1) \in \Theta(n^2)$$

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i)
    for (int j = i; j<n; ++j)
      op();
}
```

How often is `op()` called as a function of n?

$$\sum_{i=1}^{n-1}\sum_{j=i}^{n-1} 1 = \sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \in \Theta(n^2)$$

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i){
    op();
    for (int j = i; j<n; ++j)
      op();
  }
}
```

How often is `op()` called?

$$\sum_{i=1}^{n-1}\left(1+\sum_{j=i}^{n-1} 1\right) = \sum_{i=1}^{n-1}(1+(n-i)) = n-1+\frac{n(n-1)}{2} \in \Theta(n^2)$$

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i){
    op();
    for (int j = 1; j<i*i; ++j)
      op();
  }
}
```

How often is `op()` called?

$$\sum_{i=1}^{n-1}\left(1+\sum_{j=1}^{i^2-1} 1\right) = \sum_{i=1}^{n-1}\left(1+i^2-1\right) = \sum_{i=1}^{n-1} i^2 \in \Theta(n^3)$$

Asymptotic Running Times with Θ

```
void run(int n){
  for(int i = 1; i <= n; ++i)
    for(int j = 1; j*j <= n; ++j)
      for(int k = n; k >= 2; --k)
        op();
}
```

How often is `op()` called as a function of n?

$$\sum_{i=1}^{n} \sum_{j=1}^{\lfloor \sqrt{n} \rfloor} n - 1 \in \Theta\left(\sum_{i=1}^{n} n^{3/2}\right) = \Theta(\sqrt{n^5})$$

Asymptotic Running Times with Θ

```
int f(int n){
  i=1;
  while (i <= n*n*n){
    i = i*2;
    op();
  }
  return i;
}
```

How often is `op()` called as a function of n?

$$|i \in \mathbb{N} : 2^i \leq n^3| \in \Theta(\log_2 n^3) = \Theta(\log n)$$

5. Appendix

Some formulas with derivation

Sums

$$\sum_{i=0}^{n} i = \frac{n \cdot (n+1)}{2}$$

Why?

Intuition

$$1 + ... + 100 = (1 + 100) + (2 + 99) + (3 + 98) + ... + (50 + 51)$$

More formally?

Sums

$$\sum_{i=0}^{n}(n-i) = \sum_{i=0}^{n} i$$

$$\Rightarrow 2 \cdot \sum_{i=0}^{n} i = \sum_{i=0}^{n} i + \sum_{i=0}^{n}(n-i)$$
$$= \sum_{i=0}^{n}(i + (n-i)) = \sum_{i=0}^{n} n = (n+1) \cdot n$$

Sums

$$\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

This you do not need to know by heart. But you should know that it is a polynomial of third degree.

Sums

How do you derive something like this? Interesting Trick: On the one hand

$$\sum_{i=0}^{n} i^3 - \sum_{i=1}^{n} (i-1)^3 = \sum_{i=0}^{n} i^3 - \sum_{i=0}^{n-1} i^3 = n^3,$$

on the other hand

$$\begin{aligned}\sum_{i=0}^{n} i^3 - \sum_{i=1}^{n} (i-1)^3 &= \sum_{i=1}^{n} i^3 - \sum_{i=1}^{n} (i-1)^3 \\ &= \sum_{i=1}^{n} i^3 - (i-1)^3 = \sum_{i=1}^{n} 3 \cdot i^2 - 3 \cdot i + 1\end{aligned}$$

Exponents and Logarithms

$$\log_a y = x \Leftrightarrow a^x = y \quad (a > 0, y > 0)$$

$$a^x \cdot a^y = a^{x+y}$$

$$\frac{a^x}{a^y} = a^{x-y}$$

$$a^{x \cdot y} = (a^x)^y$$

$$\log_a(x \cdot y) = \log_a x + \log_a y$$

$$\log_a \frac{x}{y} = \log_a x - \log_a y$$

$$\log_a x^y = y \log_a x$$

$$\log_a n! = \sum_{i=1}^{n} \log i$$

$$\log_b x = \log_b a \cdot \log_a x$$

$$a^{\log_b x} = x^{\log_b a}$$

To see the last line, replace $x \to a^{\log_a x}$

Comparisons

$$\frac{n^2}{2^n} \underset{n \to \infty}{\longrightarrow} 0$$

Comparisons

$$\frac{n^{10000}}{2^n} \underset{n\to\infty}{\longrightarrow} 0$$

Comparisons

$d > 1, c > 0$

$$\frac{n^c}{d^n} \xrightarrow[n \to \infty]{} 0$$

because

$$\frac{n^c}{d^n} = \frac{2^{\log_2 n^c}}{2^{\log_2 d^n}} = 2^{c \cdot \log_2 n - n \log_2 d}$$

Comparisons

$$\frac{n}{\log n} \xrightarrow[n\to\infty]{} \infty$$

Comparisons

$$\frac{n \log n}{\sqrt{n}} \xrightarrow[n \to \infty]{} \infty$$

Comparisons

$$\frac{\log_2 n^2}{\sqrt{n}} \xrightarrow[n\to\infty]{} 0$$

$$\begin{aligned}\log_2 n^2 &= 2\log_2 n\\ \sqrt{n} &= n^{1/2} = 2^{\log_2 n^{1/2}} = \left(\sqrt{2}\right)^{\log_2 n}\\ \frac{\log n^2}{\sqrt{n}} &= 2\frac{\log_2 n}{\left(\sqrt{2}\right)^{\log_2 n}}\end{aligned}$$

which behaves because of $\log_2 n \to \infty$ for $n \to \infty$ like $2\frac{n}{\left(\sqrt{2}\right)^n}$

6. Tips for **code** expert

Tips for **code** expert Exercise 1

All Text Tasks

- Please learn a little LaTeX and Markdown. It will make your (and my) life a lot easier
- Useful Links and some tools I use
  - Just Enough LaTeX to Survive - Videos
  - Just Enough LaTeX to Survive - PDF
  - Detexify (OCR for LaTeX)
  - Mathpix Snipping Tool (paid)
  - Online Markdown Tutorial
  - Another Online Markdown Tutorial
  - Markdown Renderers Overview
  - Overleaf Tutorial
  - Overleaf via ETH

Task "Some Proofs"

- No need for a rigorous proof (this is not Disk Math)
- It pays off to revisit some of the $\log$-properties that we've covered today

Tips for **code** expert Exercise 2

Task "Prefix Sum in 2D"

- Study the Prefix Sum in 1D well and go from there
- Make sketches!

Task "Sliding Window"

- Sketches!

Task "Proofs by Induction"

- The binomial formula will be useful for the second one

Task "Karatsuba Ofman"

- Just translate the math into code

7. Outro

General Questions?

See you next time

Have a nice week!