

# Algorithmen und Datenstrukturen

## A9. Laufzeitanalyse: Landau-Symbole

Marcel Lüthi and Gabriele Röger

Universität Basel

15. März 2023

# Algorithmen und Datenstrukturen

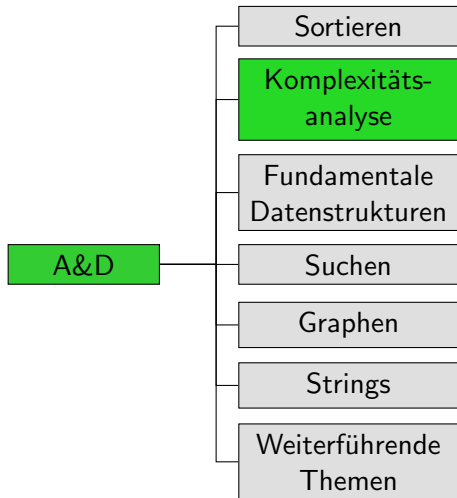
15. März 2023 — A9. Laufzeitanalyse: Landau-Symbole

## A9.1 Landau-Notation

## A9.2 Rechenregeln

## A9.3 Zusammenfassung

# Inhalt dieser Veranstaltung



# A9.1 Landau-Notation

# Ergebnis für Mergesort

„Die Laufzeit von Mergesort wächst genauso schnell wie  $n \log_2 n$ .“

## Theorem

Mergesort hat *leicht überlineare Laufzeit*, d.h.

es gibt Konstanten  $c, c', n_0 > 0$ , so dass für alle  $n \geq n_0$ :

$$cn \log_2 n \leq T(n) \leq c'n \log_2 n.$$

- ▶ Wir haben Terme niedrigerer Ordnung (Konstanten und  $n$ ) in der Abschätzung ignoriert bzw. verschwinden lassen.
- ▶ Wir haben uns nicht für die genauen Werte der Konstanten interessiert, es reicht, wenn irgendwelche passenden Konstanten existieren.
- ▶ Die Laufzeit für kleine  $n$  ist nicht so wichtig.

## Mehr bisherige Ergebnisse

### Theorem

Der Merge-Schritt hat *lineare Laufzeit*, d.h. es gibt Konstanten  $c, c', n_0 > 0$ , so dass für alle  $n \geq n_0$ :  $cn \leq T(n) \leq c'n$ .

### Theorem

Mergesort hat *leicht überlineare Laufzeit*, d.h. es gibt Konstanten  $c, c', n_0 > 0$ , so dass für alle  $n \geq n_0$ :  $cn \log_2 n \leq T(n) \leq c'n \log_2 n$ .

### Theorem

Selectionsort hat *quadratische Laufzeit*, d.h. es gibt Konstanten  $c > 0, c' > 0, n_0 > 0$ , so dass für  $n \geq n_0$ :  $cn^2 \leq T(n) \leq c'n^2$ .

Können wir das nicht irgendwie kompakter aufschreiben?

# Edmund Landau



## Edmund Landau

- ▶ deutscher Mathematiker (1877–1938)
- ▶ analytische Zahlentheorie
- ▶ kein Freund angewandter Mathematik

International: **Bachmann–Landau-Notation** auch nach Paul Gustav Heinrich Bachmann (deutscher Mathematiker)

# Landau-Symbol Theta

## Definition

Für eine Funktion  $g : \mathbb{N} \rightarrow \mathbb{R}$  ist  $\Theta(g)$  die Menge aller Funktionen  $f : \mathbb{N} \rightarrow \mathbb{R}$ , die **genauso schnell wachsen** wie  $g$ :

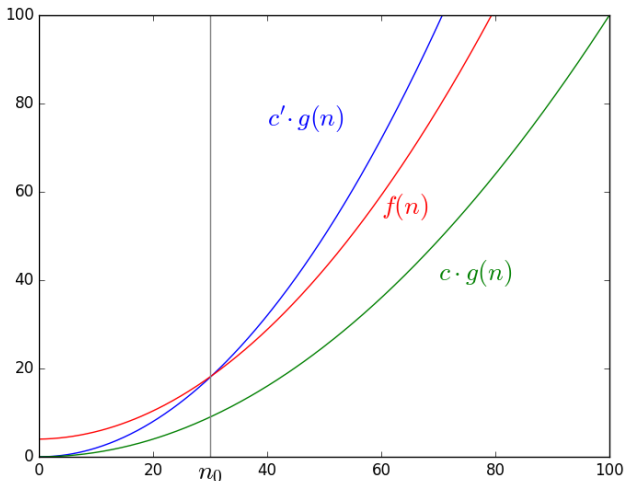
$$\Theta(g) = \{f \mid \exists c > 0 \exists c' > 0 \exists n_0 > 0 \forall n \geq n_0 : \\ c \cdot g(n) \leq f(n) \leq c' \cdot g(n)\}$$

„Die Laufzeit von Mergesort ist in  $\Theta(n \log_2 n)$ .“  
oder auch  
„Die Laufzeit von Mergesort ist  $\Theta(n \log_2 n)$ .“



# Landau-Symbol Theta: Illustration

$$f \in \Theta(g)$$



# Jupyter-Notebook (mit Aufgaben)



Jupyter-Notebook: `landau.ipynb`

# Mehr Landau-Symbole

- ▶ „ $f$  wächst nicht wesentlich schneller als  $g$ “

$$O(g) = \{f \mid \exists c > 0 \exists n_0 > 0 \forall n \geq n_0 : f(n) \leq c \cdot g(n)\}$$

- ▶  $O$  für „Ordnung“ der Funktion
- ▶ „ $f$  wächst nicht wesentlich langsamer als  $g$ “

$$\Omega(g) = \{f \mid \exists c > 0 \exists n_0 > 0 \forall n \geq n_0 : c \cdot g(n) \leq f(n)\}$$

- ▶ Es gilt  $\Theta(g) = O(g) \cap \Omega(g)$ .
- ▶ Es gilt  $f \in \Omega(g)$  gdw.  $g \in O(f)$ .
- ▶ In der Informatik interessieren wir uns oft nur für die Begrenzung des Laufzeitwachstums nach oben:  $O$  statt  $\Theta$

**Aussprache:**  $\Theta$ : Theta,  $\Omega$ : Omega,  $O$ : Oh

# Seltener benötigte Landau-Symbole

- ▶ „ $f$  wächst langsamer als  $g$ “

$$o(g) = \{f \mid \forall c > 0 \exists n_0 > 0 \forall n \geq n_0 : f(n) \leq c \cdot g(n)\}$$

- ▶ „ $f$  wächst schneller als  $g$ “

$$\omega(g) = \{f \mid \forall c > 0 \exists n_0 > 0 \forall n \geq n_0 : c \cdot g(n) \leq f(n)\}$$

Aussprache:  $\omega$ : kleines Omega

# Interessante Funktionsklassen

In aufsteigender Ordnung (abgesehen von allgemeinen  $n^k$ ):

$g$	Wachstum
1	konstant
$\log n$	logarithmisch
$n$	linear
$n \log n$	leicht überlinear
$n^2$	quadratisch
$n^3$	kubisch
$n^k$	polynomiell (Konstante $k$ )
$2^n$	exponentiell

**jwcarroll**

@jwcarroll

Folgen



## Alternative Big O notation:

 $O(1) = O(\text{yeah})$  $O(\log n) = O(\text{nice})$  $O(n) = O(\text{ok})$  $O(n^2) = O(\text{my})$  $O(2^n) = O(\text{no})$  $O(n!) = O(\text{mg!})$ 

10:10 - 6. Apr. 2019

**6.302** Retweets **15.739** „Gefällt mir“-Angaben

110



6,3 Tsd.



16 Tsd.



## A9.2 Rechenregeln

# Beispiele $\Theta$

- ▶ Bei der Analyse interessiert nur der Term höchster Ordnung (= am schnellsten wachsender Summand) einer Funktion.
- ▶ Beispiele
  - ▶  $f_1(n) = 5n^2 + 3n - 9 \in \Theta(n^2)$
  - ▶  $f_2(n) = 3n \log_2 n + 2n^2 \in \Theta(n^2)$
  - ▶  $f_3(n) = 9n \log_2 n + n + 17 \in \Theta(n \log n)$
  - ▶  $f_4(n) = 8 \in \Theta(1)$



# Beispiele Gross-O

- ▶ Bei der Analyse interessiert nur der Term höchster Ordnung (= am schnellsten wachsender Summand) einer Funktion.
- ▶ Beispiele
  - ▶  $f_1(n) = 8n^2 - 3n - 9 \in O(n^2)$
  - ▶  $f_2(n) = n^3 - 3n \log_2 n \in O(n^3)$
  - ▶  $f_3(n) = 3n \log_2 n + 1000n + 10^{200} \in O(n \log n)$
- ▶ Warum ist das so?

# Zusammenhänge

Es gilt:

- ▶  $O(1) \subset O(\log n) \subset O(n) \subset O(n \log n) \subset O(n^k) \subset O(2^n)$   
(für  $k \geq 2$ )
- ▶  $O(n^{k_1}) \subset O(n^{k_2})$  für  $k_1 < k_2$   
z.B.  $O(n^2) \subset O(n^3)$

# Rechenregeln

▶ **Produkt**

$$f_1 \in O(g_1) \text{ und } f_2 \in O(g_2) \Rightarrow f_1 f_2 \in O(g_1 g_2)$$

▶ **Summe**

$$f_1 \in O(g_1) \text{ und } f_2 \in O(g_2) \Rightarrow f_1 + f_2 \in O(g_1 + g_2)$$

▶ **Multiplikation mit Konstante**

$$k > 0 \text{ und } f \in O(g) \Rightarrow kf \in O(g)$$

$$k > 0 \Rightarrow O(kg) = O(g)$$

# Grund für Beschränkung auf Term höchster Ordnung

Beispiel:  $5n^3 + 2n \in O(n^3)$

- ▶ Wegen Regel bzgl. Multiplikation mit Konstante:
  - ▶  $5n^3 \in O(n^3)$
  - ▶  $2n \in O(n)$
- ▶ Wegen  $O(n) \subset O(n^3)$  und  $2n \in O(n)$ :
  - ▶  $2n \in O(n^3)$
- ▶ Wegen Summenregel:
  - ▶  $5n^3 + 2n \in O(n^3 + n^3)$
- ▶ Mit Multiplikation mit Konstante (bei Klasse):
  - ▶  $5n^3 + 2n \in O(n^3)$

## A9.3 Zusammenfassung

# Zusammenfassung

- ▶ Mit **Landau-Symbolen** definiert man Klassen von Funktionen, die nicht schneller/nicht langsamer/... **wachsen als eine Funktion  $g$** .
  - ▶  $O(g)$ : Wachstum nicht schneller als  $g$
  - ▶  $\Theta(g)$ : Wachstum im Wesentlichen wie  $g$