

# Algorithmen & Datenstrukturen

Sebastian Wild  
wild@cs.uni-kl.de



*Fachbereich Informatik*

10. Januar 2017

# 3

## Analyse von Algorithmen



Aufgaben zur Vorbereitung  
Warum Algorithmen analysieren?  
Was ist ein *performance bug*?

- Abstrahiert von unwichtigen Details



- Abstrahiert von unwichtigen Details  
für Gesamtlaufzeit





# Warum Asymptotik?

- Abstrahiert von unwichtigen Details  
für Gesamtlaufzeit

*“Premature optimization is the root of all evil”*

Quote by C.A.R. 'Tony' Hoare, popularized by Donald Knuth

# Warum Asymptotik?

- Abstrahiert von unwichtigen Details
- Vereinfacht die mathematische Analyse

für Gesamtlaufzeit

*“Premature optimization is the root of all evil”*

Quote by C.A.R. 'Tony' Hoare, popularized by Donald Knuth

# Warum Asymptotik?

- Abstrahiert von unwichtigen Details  
    für Gesamtlaufzeit
- Vereinfacht die mathematische Analyse
- Erlaubt oft erst sinnvollen Vergleich von Algorithmen

*“Premature optimization is the root of all evil”*

Quote by C.A.R. 'Tony' Hoare, popularized by Donald Knuth



# Laufzeit vs. Modelle



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

~ Laufzeit-Experimente

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

≠ *universal truths*

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...
- ≠ *universal truths*
- **mathematische Modelle** für Kosten



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...
- ≠ *universal truths*
- **mathematische Modelle** für Kosten
  - $\approx$  Laufzeit auf RAM (Anzahl Schritte)



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...
- ≠ *universal truths*
- **mathematische Modelle** für Kosten
  - $\approx$  Laufzeit auf RAM (Anzahl Schritte)
  - gilt (ungefähr) für alle Computer

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...
- ≠ *universal truths*
- **mathematische Modelle** für Kosten
  - $\approx$  Laufzeit auf RAM (Anzahl Schritte)
  - gilt (ungefähr) für alle Computer
  - typisch: Beschränkung auf *dominante* Operationen ("inner loop")

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

≠ *universal truths*

- **mathematische Modelle** für Kosten
  - $\approx$  Laufzeit auf RAM (Anzahl Schritte)
  - gilt (ungefähr) für alle Computer z.B. #array accesses
  - typisch: Beschränkung auf *dominante* Operationen ("inner loop")

- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ↪ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

≠ *universal truths*

- **mathematische Modelle** für Kosten

- $\approx$  Laufzeit auf RAM (Anzahl Schritte)
- gilt (ungefähr) für alle Computer z.B. #array accesses
- typisch: Beschränkung auf *dominante* Operationen ("inner loop")

↪  $\sim a \cdot (\text{Laufzeit auf RAM})$  für Konstante  $a$



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

≠ *universal truths*

- **mathematische Modelle** für Kosten

- $\approx$  Laufzeit auf RAM (Anzahl Schritte)
- gilt (ungefähr) für alle Computer z.B. #array accesses
- typisch: Beschränkung auf *dominante* Operationen ("inner loop")

~  $\sim a \cdot (\text{Laufzeit auf RAM})$  für Konstante  $a$

- mathematische Analyse (im Prinzip) möglich



- Laufzeiten (auf echter Maschine) sind **chaotisch**  
zu viele Faktoren für exakte Analyse

## ~ Laufzeit-Experimente

- Erkenntnisse nur für **eine** Maschine/Plattform gültig
- nur für getestete Eingaben
- nur für getestete Implementierung
- ...

≠ *universal truths*

## • mathematische Modelle für Kosten

- $\approx$  Laufzeit auf RAM (Anzahl Schritte)
- gilt (ungefähr) für alle Computer
- typisch: Beschränkung auf *dominante* Operationen ("inner loop")

z.B. `#array accesses`

~  $\sim a \cdot (\text{Laufzeit auf RAM})$  für Konstante  $a$

- mathematische Analyse (im Prinzip) möglich

≠ einfach ...  
Beschäftigt AofA-Forscher-  
Community seit Jahrzehnten!

Grenzwerte tilde notation  $O$ ,  $\Omega$ ,  $\Theta$  Intuition (tilde ohne  $c$ ) künstliches Gegenbeispiel (nicht-konstanter Faktor) gibt es netten Algorithmus mit Fluktuationen im führenden Term? formale Definition  $\limsup$   $n \rightarrow \infty$  Definition sieht einfacher aus, aber technisch oft ärgerlich (negative Funktionen, nur Integer) äquivalent klein  $o$  etc

