

Offene Vorlesung / Wiederholung

Damien Foucard, Christoph Seifert | Open Distributed Systems | Einführung in die Programmierung
Wintersemester 2019/2020

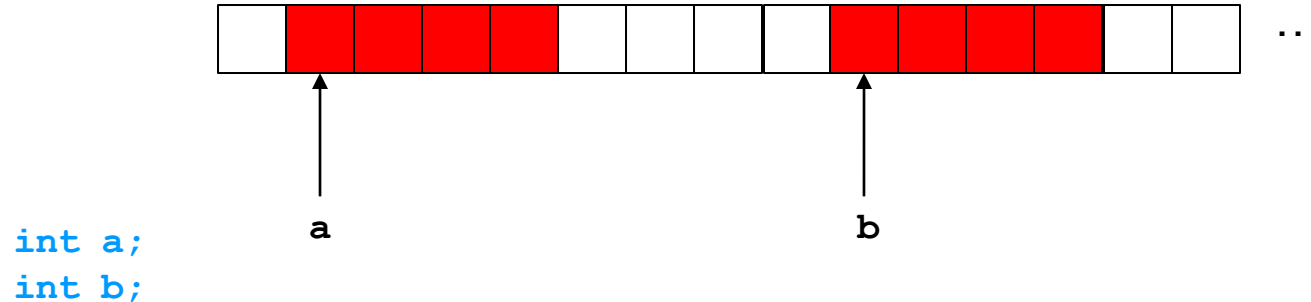
Pointer

„Because they, quite honestly, can cause electric shocks to come up through the keyboard and physically weld your arms permanently in place, cursing you to a life at the keyboard.”,

Beej's Guide to C Programming

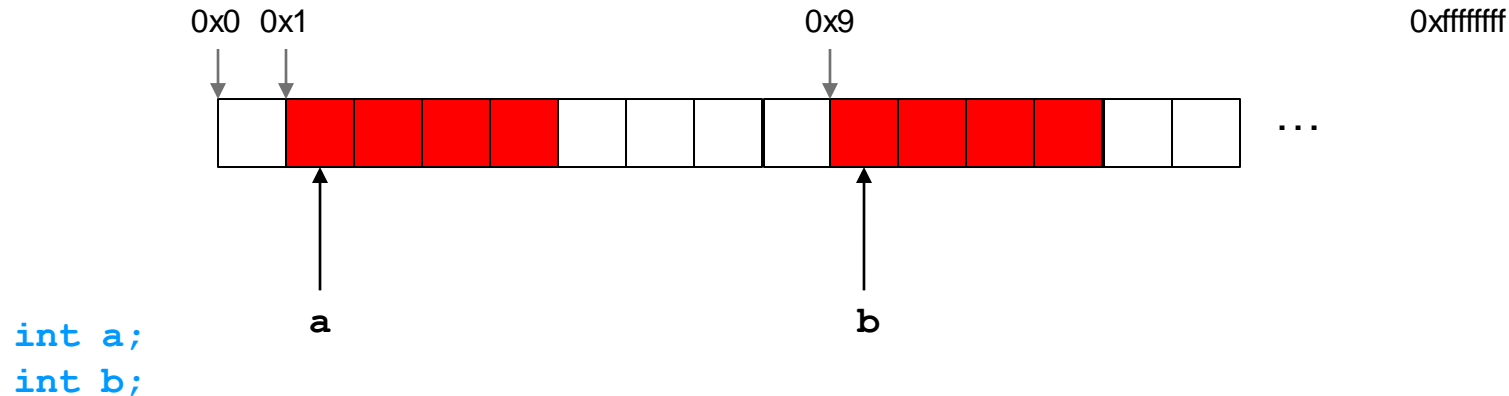
Virtueller Speicher

- “Array von Bytes”
- Linearer Speicher, initial unfragmentiert



Speicherreferenz

- Position/Adresse im Speicher → Pointer



Aber warum?

- Call-by-Value, Call-by-Reference
 - Speicherverbrauch
 - Overhead bei Allokierung von lokalen Variablen
- Pointerarithmetik
 - Freier Zugriff auf Speicherbereich (z.B. Zeichenketten)

Asymptotisches Laufzeitverhalten

Laufzeiten

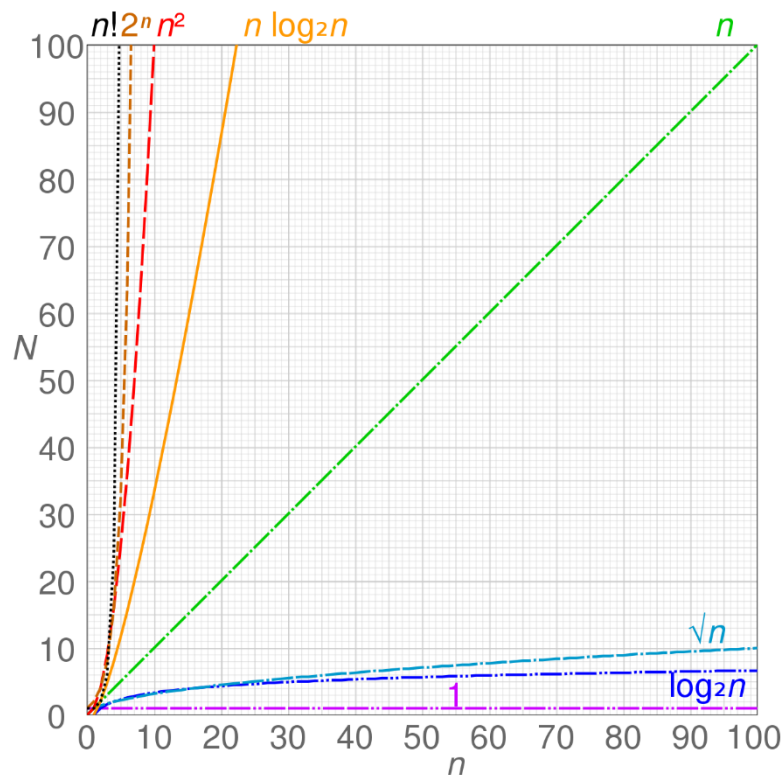


Bild: CC BY-SA 4.0,
[https://en.wikipedia.org/wiki/Computational_](https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations)
[complexity_of_mathematical_operations](https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations)

Groß-O (obere Schranke)

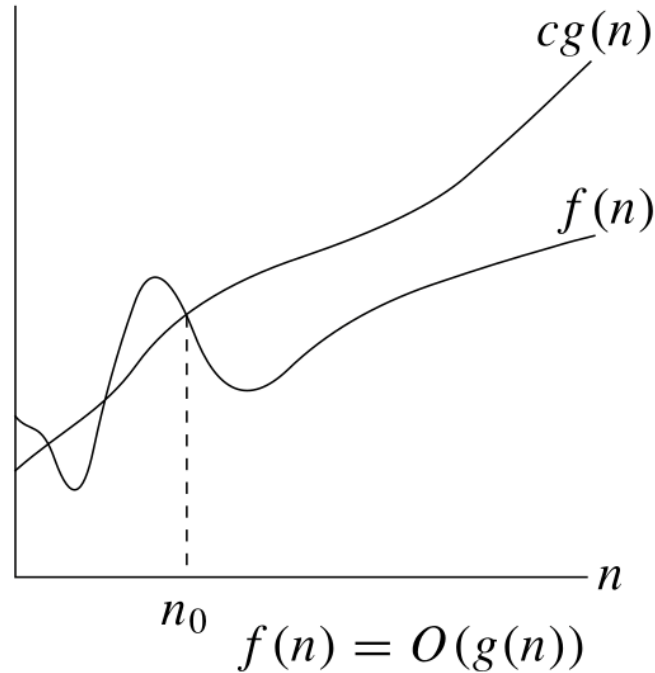


Bild: Cormen et al., 2001, Introduction to Algorithms

Groß-Omega: Ω (untere Schranke)

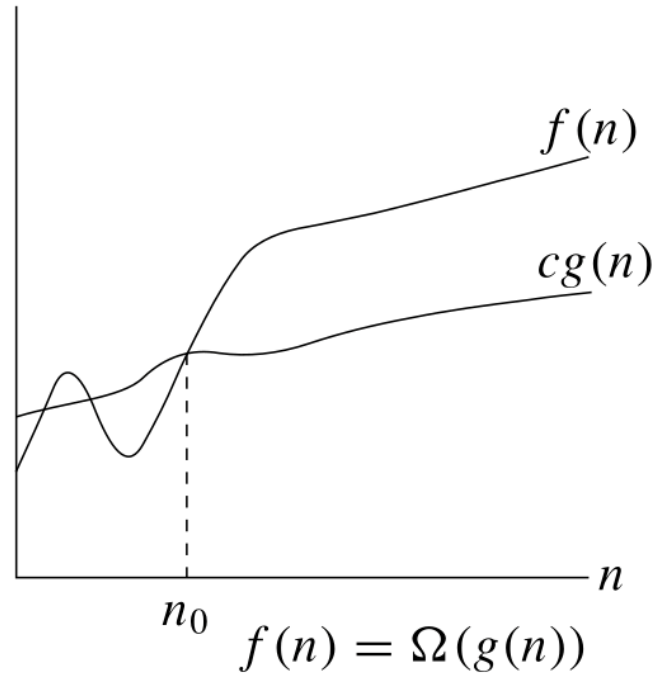


Bild: Cormen et al., 2001, Introduction to Algorithms

Groß-Theta: Θ (obere u. untere S.)

$$g(n) \in \Theta(f(n)) \Leftrightarrow \\ g(n) = O(f(n)) \text{ und } g(n) = \Omega(f(n))$$

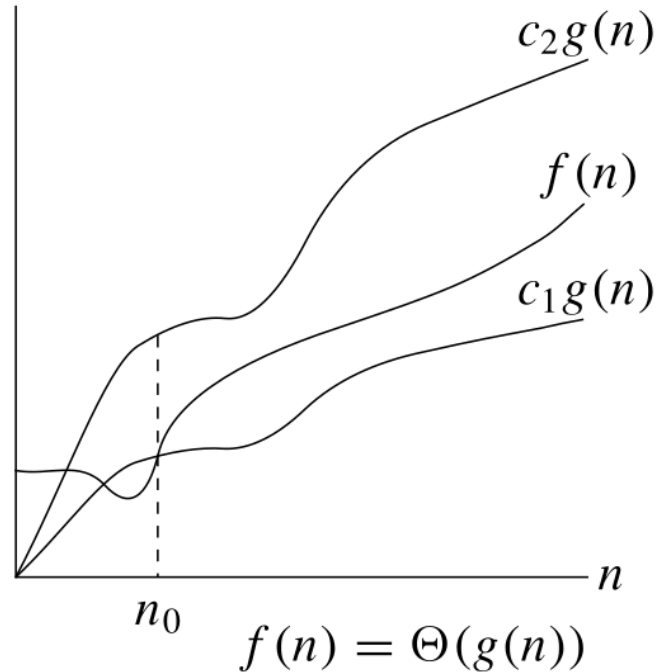


Bild: Cormen et al., 2001, Introduction to Algorithms

Asymp. Not.: Vergleich

Vereinfacht: O , Ω , Θ , o , ω sind die „asymptotischen Versionen“ von \leq , \geq , $=$, $<$, $>$ (in dieser Reihenfolge)

$f \in o(g)$	Wachstum von f	$<$	Wachstum von g
$f \in O(g)$	Wachstum von f	\leq	Wachstum von g
$f \in \Theta(g)$	Wachstum von f	$=$	Wachstum von g
$f \in \Omega(g)$	Wachstum von f	\geq	Wachstum von g
$f \in \omega(g)$	Wachstum von f	$>$	Wachstum von g

Beispiele

Insertionsort:

- Genereller Fall: $O(n^2)$, $\Omega(n) \rightarrow \Theta(?)$
- Best-Case: $O(n)$, $\Omega(n) \rightarrow \Theta(n)$

Mergesort:

- Genereller Fall: $O(n \log n)$, $\Omega(n \log n) \rightarrow \Theta(n \log n)$

Sortieren Laufzeitvergleich

Parameter

- Wertebereich m
 - $m = \mathbb{N}$,
 - $m = \{1, 2, \dots, 10\}$
- Problemegröße n
 - $n \rightarrow \infty$ (asympt. Verhalten)
 - $n = 10$
- Verteilung der Probe
 - zufällig, vorsortiert (auf- oder absteigend)

Beispiele

$m = \mathbb{N}$, $n = \{1, \dots, 10\}$, keine Annahme über Verteilung:

- Countsort, Mergesort, Insertionsort
- **Insertionsort!**

$m = \mathbb{N}$, $n = \{1, \dots, 10^6\}$, aufsteigend vorsortiert:

- Quicksort, Mergesort, Insertionsort
- **Insertionsort!**