Grundzüge der Informatik 1

Vorlesung 8

Überblick

Überblick

- Wiederholung
  - Integer Multiplikation
  - Matrix Multiplikation (Anfang)
- Matrix Multiplikation
  - Verbesserter Algorithmus
- Auflösen von Rekursionsgleichungen
  - „Master Theorem“

Universität zu Köln

Teile & Herrsche Algorithmen

Wodurch unterscheiden sich Teile & Herrsche Algorithmen?

- Die Anzahl der Teilprobleme
- Die Größe der Teilprobleme
- Den Algorithmus für das Zusammensetzen der Teilprobleme
- Den Rekursionsabbruch

Wann lohnt sich Teile & Herrsche?

- Kann durch Laufzeitanalyse vorhergesagt werden

Laufzeitanalyse
- Rekursionen

Laufzeiten der Form

$$T(n) = a\,T(n/b) + f(n)$$

(und $T(1) = O(1)$)

Laufzeitanalyse
- Rekursionen

Laufzeiten der Form

$$T(n) = a\,T(n/b) + f(n)$$

(und $T(1) = O(1)$)

Laufzeitanalyse
- Rekursionen

Laufzeiten der Form

$$T(n) = a\, T(n/b) + f(n)$$

Anzahl Unterprobleme

(und $T(1) = O(1)$)

Universität zu Köln

Laufzeitanalyse
- Rekursionen

Laufzeiten der Form

$$T(n) = a\,T(n/b) + f(n)$$

Anzahl Unterprobleme

(und $T(1) = O(1)$)

Größe der Unterprobleme
(bestimmt Höhe des Rekursionsbaums)

Universität zu Köln

Laufzeitanalyse
- Rekursionen

Laufzeiten der Form

$$T(n) = a\,T(n/b) + f(n)$$

Anzahl Unterprobleme

Aufwand für Aufteilen und Zusammenfügen

Größe der Unterprobleme
(bestimmt Höhe des Rekursionsbaums)

(und $T(1) = O(1)$)

Teile & Herrsche Prinzip
- Multiplikation großer Zahlen

Integer Multiplikation

- Problem: Multipliziere zwei n-Ziffer Integer
- Eingabe: zwei n-Ziffer Integer X und Y
- Ausgabe: 2n-Ziffer Integer Z mit Z=XY

Darstellung im Rechner

- Wir nehmen an, dass jede Ziffer eine Speicherzelle benötigt!
- Wir können zwei n-Ziffer Integer in $\Theta(n)$ Zeit addieren
- Wir können ein n-Ziffer Integer mit Zehnerpotenz 10^k in $\Theta(n+k)$ Zeit multiplizieren

Teile & Herrsche Prinzip
- Multiplikation großer Zahlen

Verbesserte Integer Multiplikation

A | B • C | D

B · D

\+ B · C

\+ A · D

\+ A · C

= Ergebnis

Berechne BC + AD
mit einer Multiplikation

Trick:
(A+B) (C+D) – AC – BD
= BC + AD

Laufzeitanalyse
- Verbesserte Integer Multiplikation

Verbesserte Integer Multiplikation

$$T(n) = 3\,T(n/2) + c\,n$$

Anzahl Unterprobleme

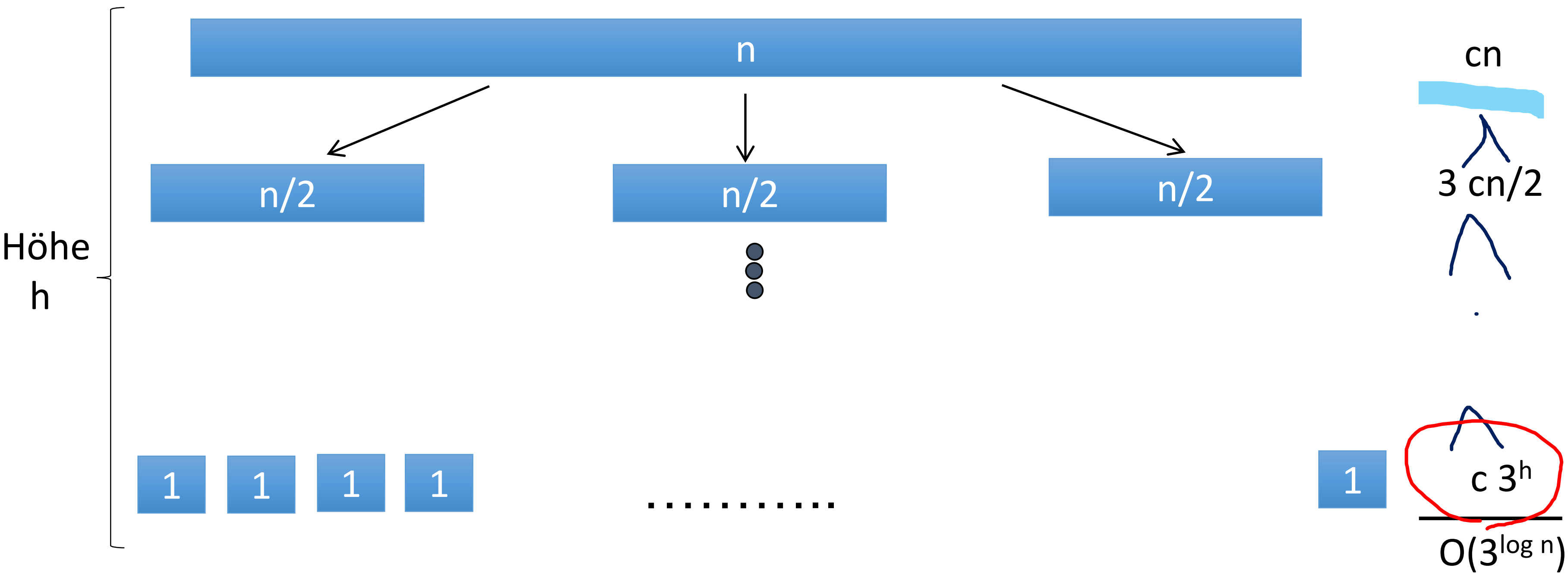
Aufwand für Aufteilen und
Zusammenfügen

Größe der Unterprobleme
(bestimmt Höhe des Rekursionsbaums)

(und $T(1) = O(1)$)

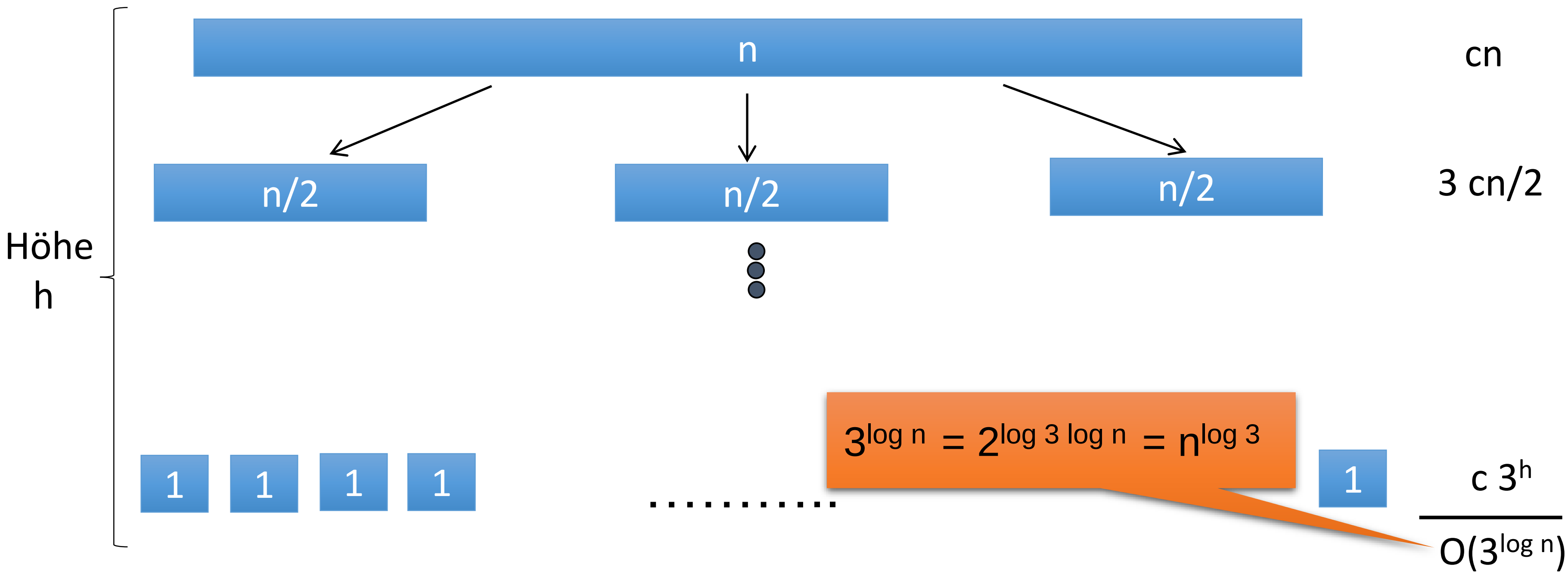
Laufzeitanalyse – grafische Darstellung - Verbesserte Integer Multiplikation

- Auflösen von $T(n) \leq 3\,T(n/2) + cn$ (Intuition)
- $T(1) = c$



Laufzeitanalyse – grafische Darstellung - Verbesserte Integer Multiplikation

- Auflösen von $T(n) \leq 3\,T(n/2) + cn$ (Intuition)
- $T(1) = c$

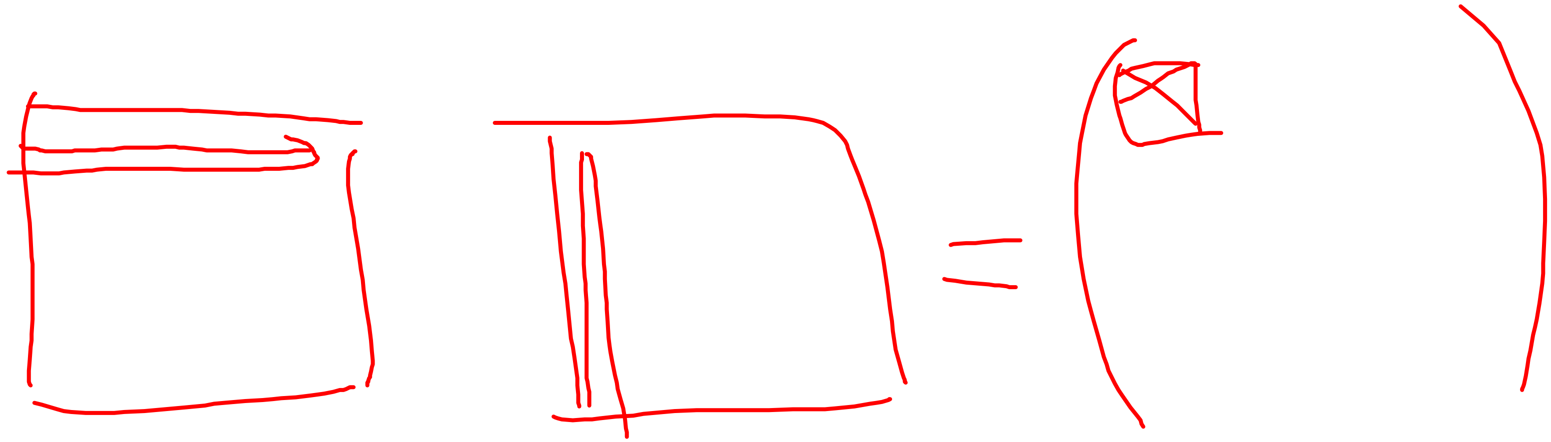


Universität zu Köln

Teile & Herrsche Prinzip
- Matrixmultiplikation

Matrixmultiplikation

- Problem: Berechne das Produkt zweier n×n-Matrizen
- Eingabe: Matrizen X, Y
- Ausgabe: Matrix Z = X · Y



Teile & Herrsche Prinzip - Matrixmultiplikation

Matrixmultiplikation

- $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{pmatrix}$

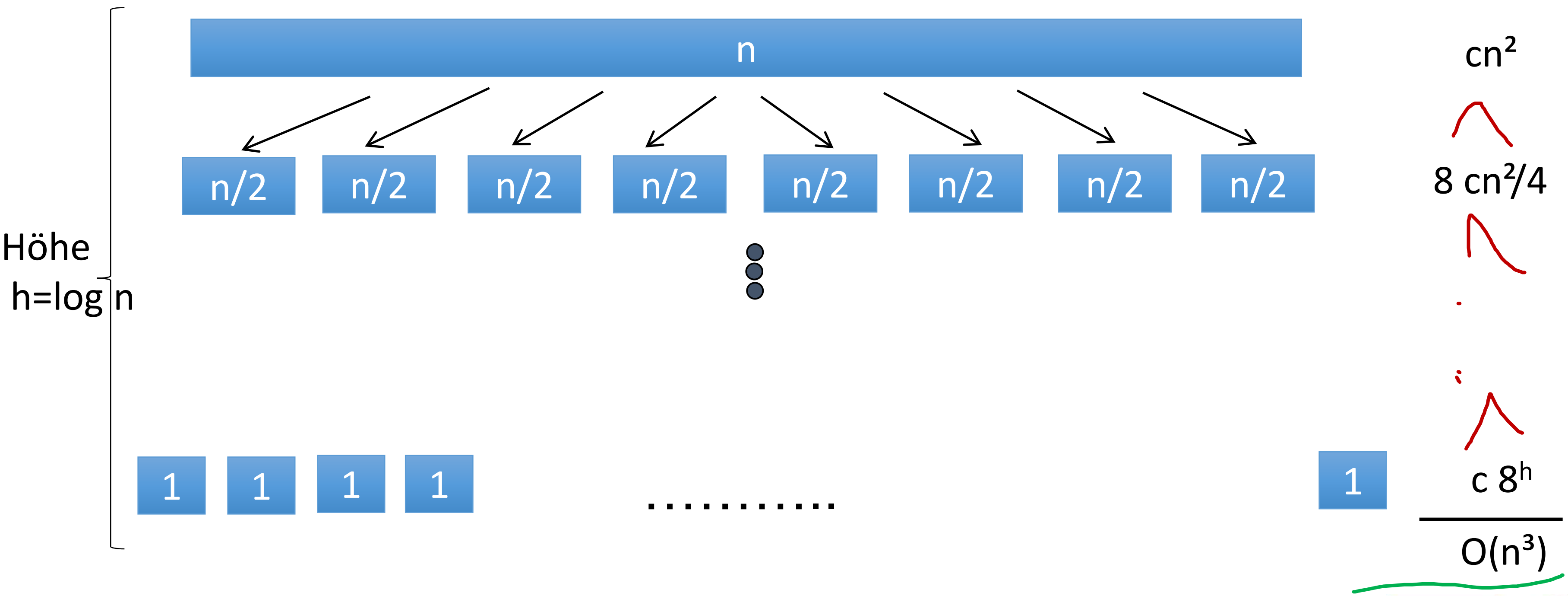
Laufzeit

- 8 Multiplikationen von n/2×n/2 Matrizen
- 4 Additionen von n/2 ×n/2 Matrizen

- $T(n) \leq \begin{cases} 8T\left(\frac{n}{2}\right) + cn^2 & \text{, für } n > 1 \\ c & \text{, für } n = 1 \end{cases}$

Teile & Herrsche Prinzip
- Matrixmultiplikation

- Auflösen von $T(n) \leq 8\, T(n/2) + cn^2$ (Intuition)
- $T(1) = c$



Teile & Herrsche Prinzip - Matrixmultiplikation

Matrixmultiplikation

- $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{pmatrix}$

Trick wie bei Integer Multiplikation

- $P_1 = A \cdot (F-H)$
- $P_2 = (A+B) \cdot H$
- $P_3 = (C+D) \cdot E$
- $P_4 = D \cdot (G-E)$
- $P_5 = (A+D) \cdot (E+H)$
- $P_6 = (B-D) \cdot (G+H)$
- $P_7 = (A-C) \cdot (E+F)$

$AE + BG = P_4 + P_5 + P_6 - P_2$
$AF + BH = P_1 + P_2$
$CE + DG = P_3 + P_4$
$AF + BH = P_1 + P_5 - P_3 - P_7$

- Nur 7 Multiplikationen!!!!

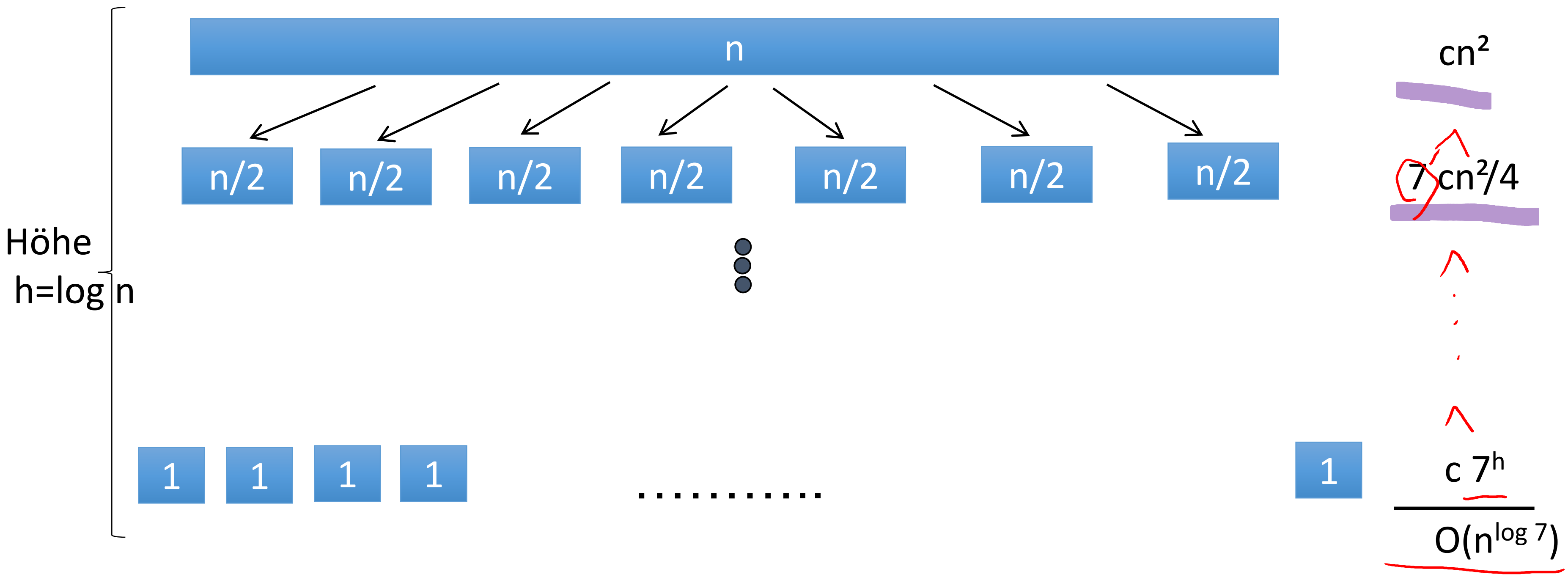
$P_4 + P_5 + P_6 - P_2 = D \cdot (G-E) + (A+D) \cdot (E+H) + (B-D) \cdot (G+H) - (A+B) \cdot H$

$= DG - DE + AE + AH + DE + DH + BG + BH - DG - DH - AH - BH$

Universität zu Köln

Teile & Herrsche Prinzip
- Matrixmultiplikation

- Auflösen von $T(n) \le 7\,T(n/2) + cn^2$ (Intuition)
- $T(1) = c$



Teile & Herrsche Prinzip
- Matrixmultiplikation

Satz 8.1 (Algorithmus von Strassen)

- Zwei $n \times n$ Matrizen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{2.81})$ Zeit multipliziert werden.

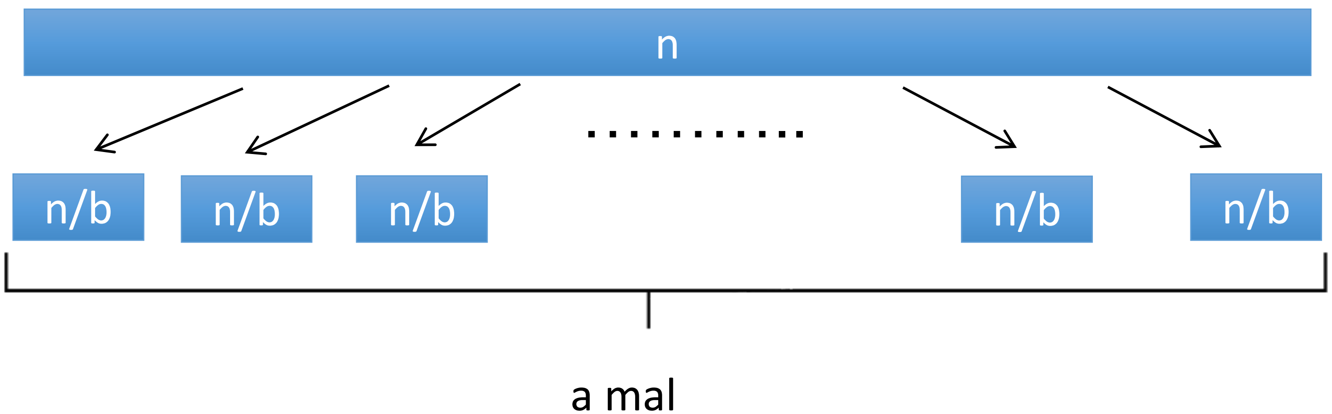
Beweis

- Laufzeit und Korrektheit können leicht per Induktion gezeigt werden.

Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Rekursionsgleichung:

- Sei T(n) = a T(n/b) + f(n) und T(1) = 1



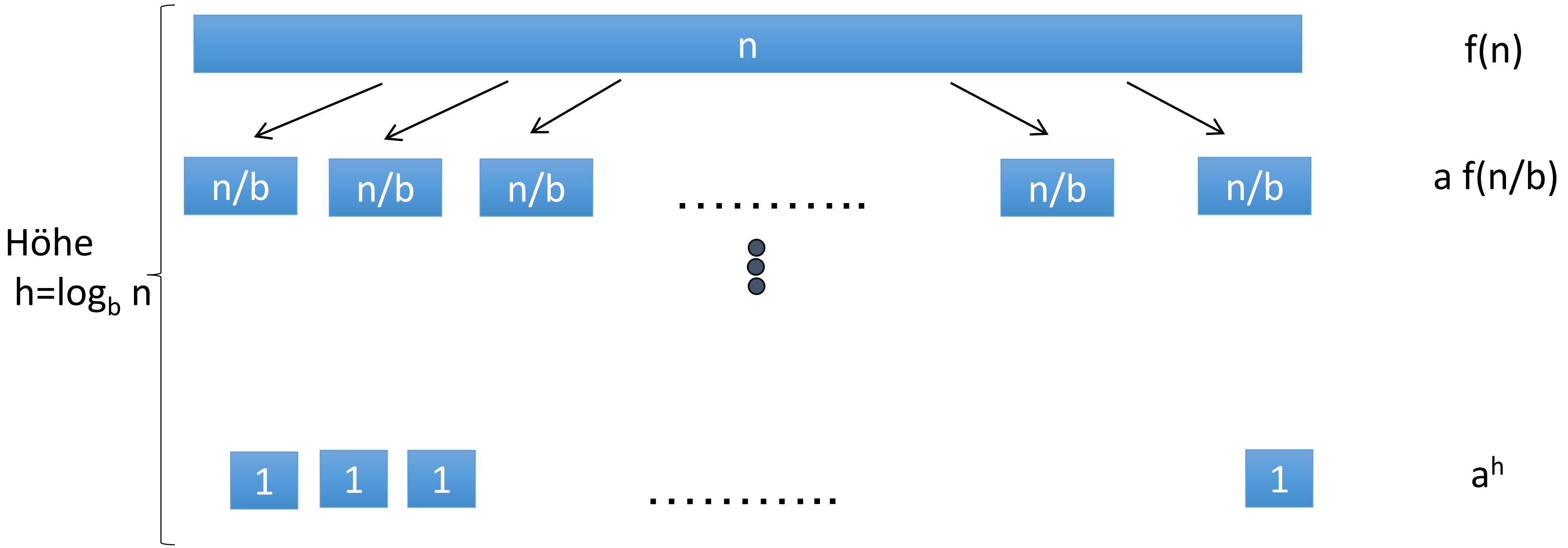
f(n)

a f(n/b)

Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Rekursionsgleichung:

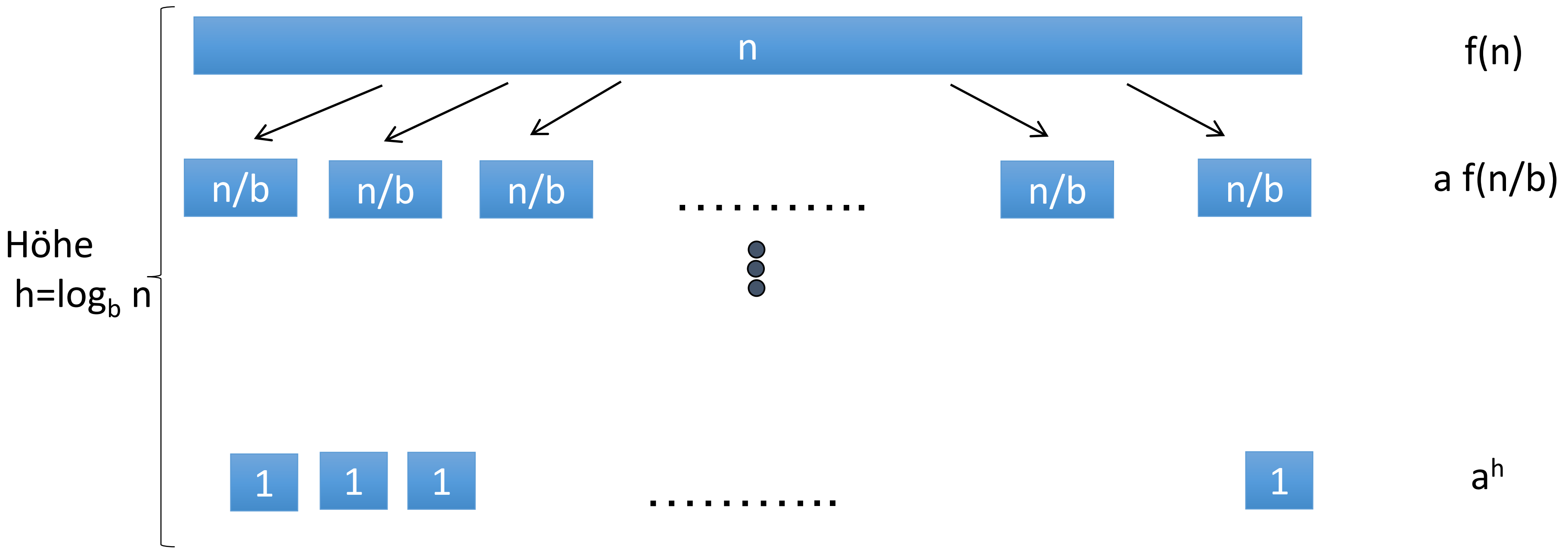
- Sei T(n) = a T(n/b) + f(n) und T(1) = 1



Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Rekursionsgleichung:

- Setze f(n) = γ a f(n/b)

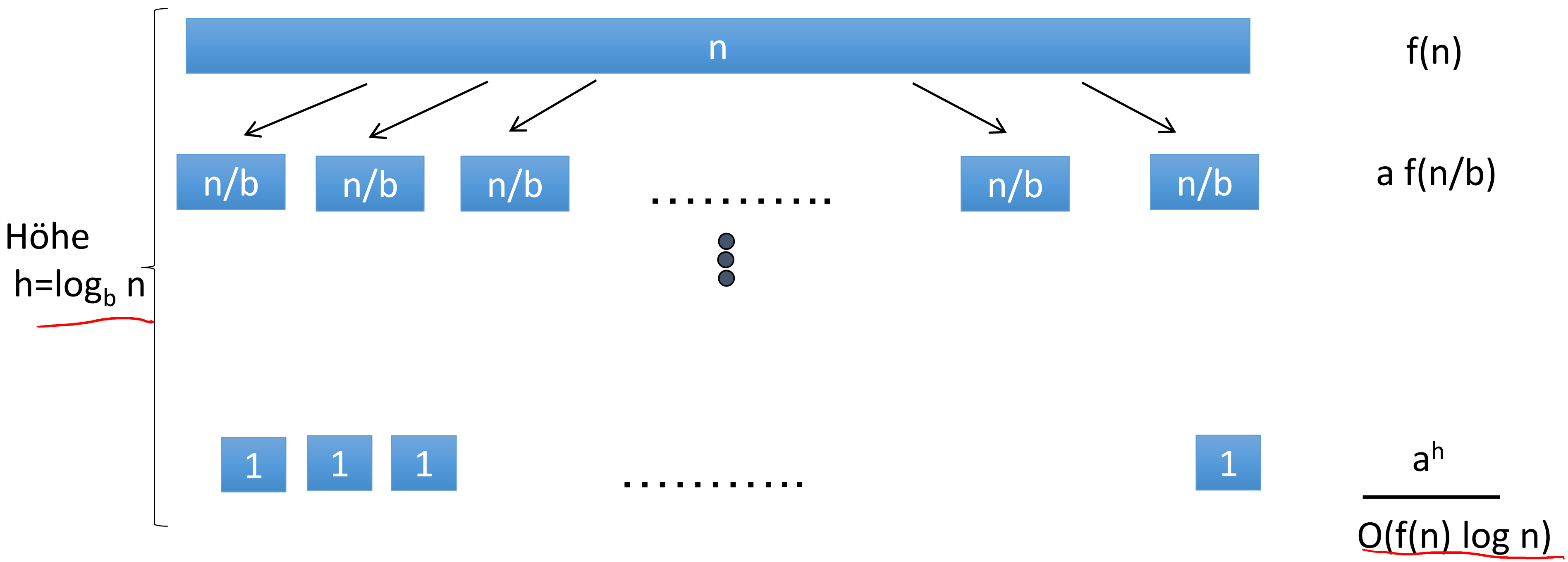


Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Fall 1:
γ=1

Rekursionsgleichung:

- Setze $f(n) = \gamma \, a \, f(n/b)$

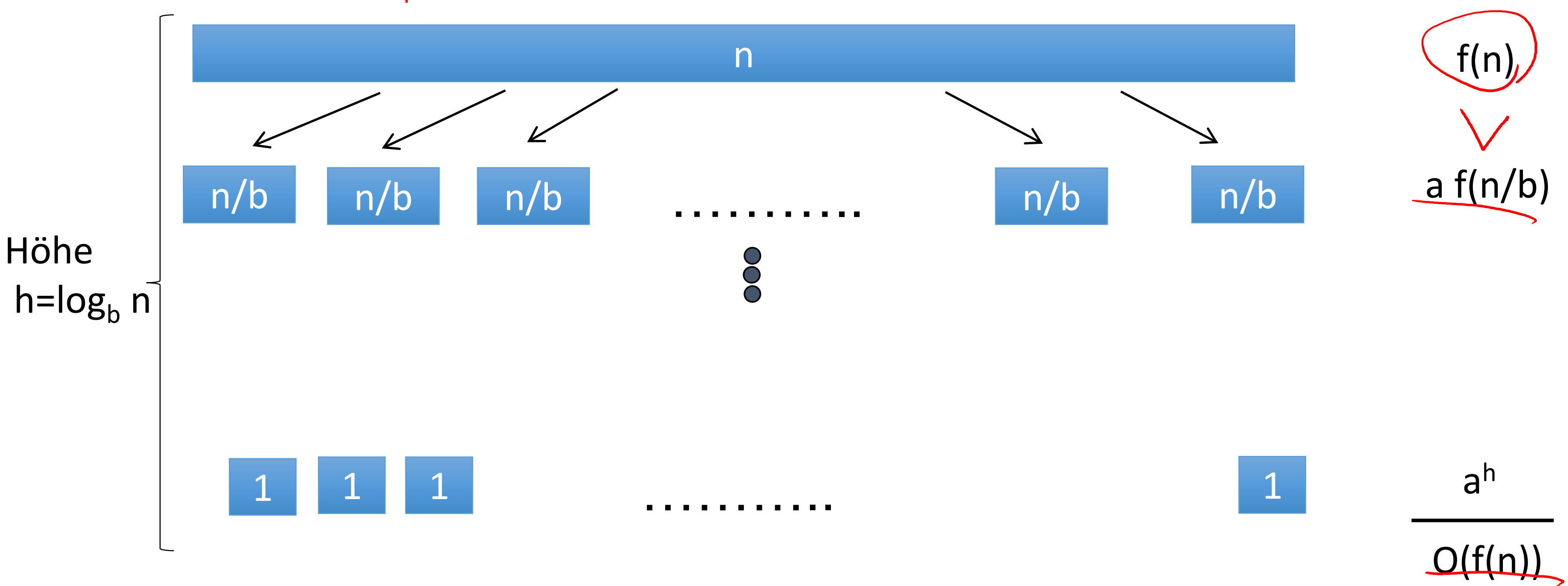


Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Fall 1:
Konstante $\gamma>1$

Rekursionsgleichung:

- Setze $f(n) = \gamma\, a\, f(n/b)$

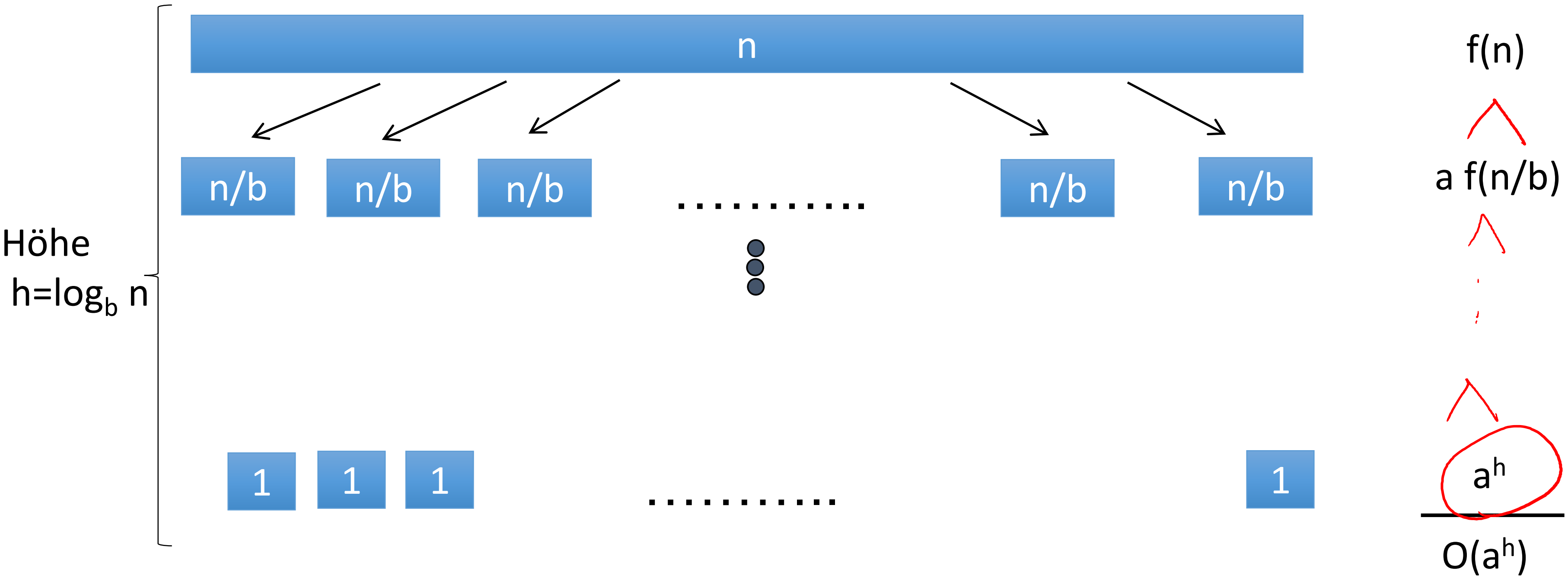


Teile & Herrsche Prinzip
- Auflösen von Rekursionsgleichungen

Fall 1:
Konstante $\gamma<1$

Rekursionsgleichung:

- Setze $f(n) = \gamma\ a\ f(n/b)$



Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Satz 8.2

- Seien a ≥ 1 und b ≥ 2 ganzzahlige Konstanten und f: $\mathbb{N}_0 \rightarrow \mathbb{N}_0$.
- Sei $T(n) \leq \begin{cases} aT\left(\frac{n}{b}\right) + f(n) & \text{, für } n > 1 \\ f(1) & \text{, für } n = 1 \end{cases}$

- Dann gilt:

(1) wenn f(n) = a f(n/b), dann ist T(n) = O(f(n) log n)

(2) wenn f(n) ≥ γ a f(n/b) für eine Konstante γ>1, dann gilt T(n) = O(f(n))

(3) wenn f(n) ≤ γ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(1) wenn f(n) = a f(n/b), dann ist T(n) = O(f(n) log n) Bsp: MergeSort

Ann.: n ist Potenz von b

z.z.: $T(n) \le f(n) \cdot (\log_b n + 1)$

Ind. Anf.: $T(1) = f(1) = f(1) \cdot (\log_b 1 + 1)$ ✓

Universität zu Köln

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(1) wenn f(n) = a f(n/b), dann ist T(n) = O(f(n) log n)

Ind. Ann.: Es gilt $T(m) \le f(m) \cdot (\log_b m + 1)$ für $m < n$, m Potenz von b

Ind. Schluss:

$$T(n) \le a \cdot T(n/b) + f(n)$$
$$\le a \cdot f(n/b) \cdot \left(\underline{\log_b n/b + 1}\right) + f(n)$$
$$\le \underbrace{a \cdot f(n/b)} \cdot \log n + f(n)$$
$$= f(n) \cdot \log n + f(n) = f(n) \cdot (\log n + 1)$$

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(2) wenn $f(n) \geq \gamma\, a\, f(n/b)$ für eine Konstante $\gamma>1$, dann gilt $T(n) = O(f(n))$

z.z.: $T(n) \le f(n) \cdot \sum_{i=0}^{\log_b n} \left(\frac{1}{\gamma}\right)^i$

= const

Ind.Anf.: $T(1) = f(1) = f(1) \cdot \sum_{i=0}^{\log_b 1} \left(\frac{1}{\gamma}\right)^i$ ✓

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(2) wenn $f(n) \geq \gamma$ a $f(n/b)$ für eine Konstante $\gamma>1$, dann gilt $T(n) = O(f(n))$

Ind. Ann.: $T(m) \le f(m) \cdot \sum_{i=0}^{\log_b m} \left(\frac{1}{\gamma}\right)^i$ gilt für

$m<n$, m Potenz von b

Ind. Schluss: $T(n) \le a \cdot T(n/b) + f(n)$

$$\le a \cdot f\left(\frac{n}{b}\right) \cdot \sum_{i=0}^{\log_b \frac{n}{b}} \left(\frac{1}{\gamma}\right)^i + f(n)$$

$$\le \frac{1}{\gamma} \cdot f(n) \cdot \sum_{i=0}^{\log_b \frac{n}{b}} \left(\frac{1}{\gamma}\right)^i + f(n)$$

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(2) wenn f(n) ≥ γ a f(n/b) für eine Konstante γ>1, dann gilt T(n) = O(f(n))

$$\frac{1}{\gamma} \cdot f(n) \cdot \sum_{i=0}^{\log_b n/b} \left(\frac{1}{\gamma}\right)^i + f(n)$$

$$= f(n) \cdot \sum_{i=1}^{\log_b n} \left(\frac{1}{\gamma}\right)^i + f(n) \cdot \left(\frac{1}{\gamma}\right)^0$$

$$= f(n) \cdot \sum_{i=0}^{\log_b n} \left(\frac{1}{\gamma}\right)^i \qquad ✓$$

Universität zu Köln

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(3) wenn f(n) $\leq \gamma$ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

zZ: $T(n) \leq a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(n) = O(a^{\log_b n})$

Ind. Anf.:

$$T(1) = f(1) = f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(1)$$

$$= a^{\log_b 1} \cdot f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(1) \quad \checkmark$$

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(3) wenn f(n) ≤ γ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

Ind. Ann.: $T(m) \le a^{\log_b m} \cdot f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(m)$

für alle m<n, m Potenz von b

Ind. Schluss: $T(n) \le a \cdot \underline{T(n/b)} + f(n)$

$\le a \cdot \left(a^{\log_b n/b} \cdot f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(n/b)\right) + f(n)$

$= a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - a \frac{\gamma}{1-\gamma} \cdot f(n/b) + f(n)$

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beweis:

(3) wenn f(n) ≤ γ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

$$a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - a \cdot \frac{\gamma}{1-\gamma} \cdot f(n/b) + f(n)$$

$$\leq a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - a \cdot \frac{1}{1-\gamma} \cdot f(n) + f(n)$$

$$\leq a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - \left(\frac{1-\gamma}{1-\gamma} + \frac{\gamma}{1-\gamma}\right) \cdot f(n) + f(n)$$

$$= a^{\log_b n} \cdot f(1) \cdot \frac{1}{1-\gamma} - \frac{\gamma}{1-\gamma} \cdot f(n)$$

✓

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Satz 8.2

- Seien a ≥ 1 und b ≥ 2 ganzzahlige Konstanten und f: $\mathbb{N}_0 \rightarrow \mathbb{N}_0$.
- Sei $T(n) \leq \begin{cases} aT\left(\frac{n}{b}\right) + f(n) & \text{, für } n > 1 \\ f(1) & \text{, für } n = 1 \end{cases}$

- Dann gilt:

  (1) wenn f(n) = a f(n/b), dann ist T(n) = O(f(n) log n)

  (2) wenn f(n) ≥ γ a f(n/b) für eine Konstante γ>1, dann gilt T(n) = O(f(n))

  (3) wenn f(n) ≤ γ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Satz 8.2

- Seien a ≥ 1 und b ≥ 2 ganzzahlige Konstanten und f: $\mathbb{N}_0 \rightarrow \mathbb{N}_0$.
- Sei $T(n) \leq \begin{cases} aT\left(\frac{n}{b}\right) + f(n) & \text{, für } n > 1 \\ f(1) & \text{, für } n = 1 \end{cases}$

$$\underline{a^{\log_b n}} = \underline{b^{\log_b a}}\, {}^{\log_b n}$$
$$= \underline{n^{\log_b a}}$$

- Dann gilt:

(1) wenn f(n) = a f(n/b), dann ist T(n) = O(f(n) log n)

(2) wenn f(n) ≥ γ a f(n/b) für eine Konstante γ>1, dann gilt T(n) = O(f(n))

(3) wenn f(n) ≤ γ a f(n/b) für eine Konstante 0<γ<1, dann gilt T(n) = O($a^{\log_b n}$)

Teile & Herrsche Prinzip
- Auflösen von Rekursionen

Beispiel:

$$T(n) = \underline{16} \cdot T(\underline{n/2}) + \underline{n^4}$$

$a = 16$

$b = 2$

$f(n) = n^4$

$f(n) \overset{?}{\substack{\le \\ \ge \\ =}} a \cdot f(n/b)$

$n^4 = 16 \cdot \left(\frac{n}{2}\right)^4 = 16 \cdot \frac{n^4}{2^4} = n^4$

Fall 1 ($\gamma = 1$):

$T(n) = O(n^4 \log n)$

Universität zu Köln

Teile & Herrsche Prinzip

Zusammenfassung

- Teile & Herrsche Prinzip:
  - Aufteilen in gleich große Teile
  - Rekursiv lösen
  - Zusammensetzen
- Teile & Herrsche Algorithmen
  - MergeSort
  - Binäre Suche
  - Integer Mutliplikation
  - Matrix Multiplikation
- Auflösen von Laufzeitrekursionen
  - „Master“-Theorem

Referenzen

- T. Cormen, C. Leisserson, R. Rivest, C. Stein. Introduction to Algorithms. The MIT press. Second edition, 2001.