

Algorithmen und Datenstrukturen

Wintersemester 2020/21

5. Vorlesung

Rekursionsgleichungen

Lösen von Rekursions(un)gleichungen

Frage: Gilt für $T(n) = 2 \cdot T(n/2) + 4n$ (mit $T(1) = 0$)
auch $T(n) \in O(n \log n)$?

Behauptung: Es gibt ein $c > 0$, so dass $T(n) \leq cn \log_2 n$.

Beweis. Durch Induktion über n . Ind.-Anfang: $T(1) \leq 0$ ✓
Induktionsannahme: $T(k) \leq ck \log_2 k$ gilt für alle $k < n$.

Wir wissen: $T(n) = 2T(n/2) + 4n$

$$\leq 2c \frac{n}{2} \log_2 \frac{n}{2} + 4n \quad (\text{wegen IA})$$

$$= cn \cdot (\log_2 n - \log_2 2) + 4n$$

$$= cn \log_2 n - cn + 4n$$

$$= cn \log_2 n + (4 - c)n$$

$$\leq cn \log_2 n \quad \text{falls } c \geq 4.$$

Substitutionsmethode:

1. Lösung von Rekursion raten
2. Mit Induktion beweisen

\Rightarrow Behauptung wahr (es gibt ein $c > 0 \dots$) $\Rightarrow T(n) \in O(n \log n)$!

Streng genommen haben wir die Behauptung nur für $n = \text{Zweierpotenz}$ bewiesen. Auf der nächsten Folie sind wir genauer.

I) Substitutionsmethode

Noch'n Beispiel: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$

Behauptung: $T(n) \in O(n)$ (mit $T(1) = 0$)

Also zeigen wir: $T(n) \leq cn$ für eine Konstante $c > 0$.

~~Beweis~~ Induktion über n .

~~IA: $T(k) \leq ck$ für alle $k < n$~~

~~Wissen: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$~~

~~$\leq c \cdot \lfloor n/2 \rfloor + c \cdot \lceil n/2 \rceil + 1$ wg. IA~~

~~$\leq c \cdot (\lfloor n/2 \rfloor + \lceil n/2 \rceil) + 1$~~

~~$\leq c \cdot n + 1$~~



Noch'n Versuch

Noch'n Beispiel: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$

Behauptung: $T(n) \in O(n)$ (mit $T(1) = 0$)

Also zeigen wir: $T(n) \leq cn + 1$ für eine Konstante $c > 0$.

~~Beweis~~ Induktion über n .

~~IA: $T(k) \leq ck + 1$ für alle $k < n$.~~

~~Wissen: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$~~

~~$$\leq (c \cdot \lfloor n/2 \rfloor + 1) + (c \cdot \lceil n/2 \rceil + 1) + 1$$~~

~~$$\leq c \cdot n + 3$$~~



Nicht verzagen!

Selbes Beispiel: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$

Behauptung: $T(n) \in O(n)$ ✓ (mit $T(1) = 0$)

Nun probieren wir: $T(n) \leq cn - d$ für Konstanten $c, d > 0$. ✓
 D.h. wir machen unsere Aussage *schärfer!!*

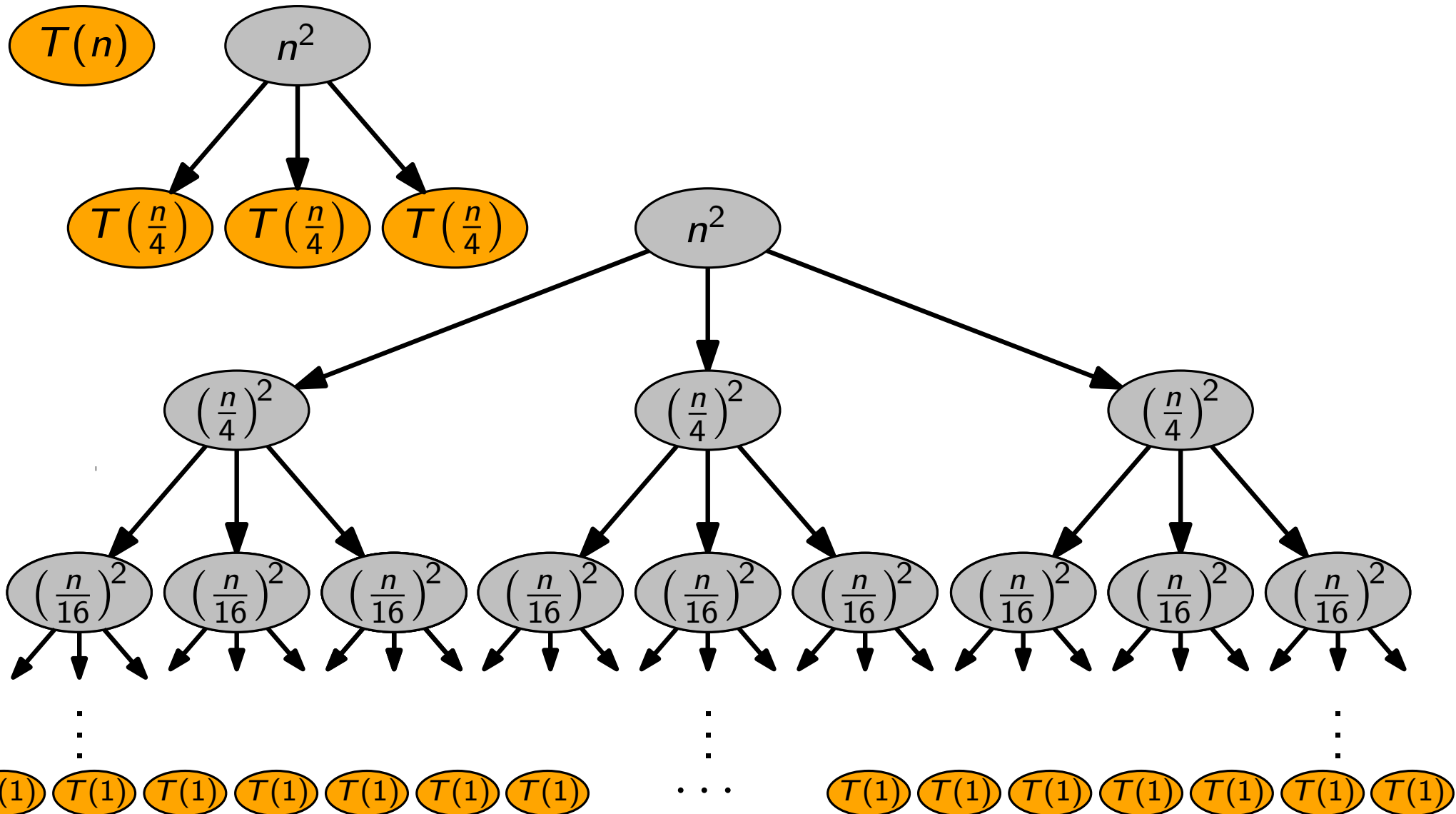
Beweis. Induktion über n .

IA: $T(k) \leq ck - d$ für alle $k < n$.

Wissen: $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$
 $\leq (c\lfloor n/2 \rfloor - d) + (c\lceil n/2 \rceil - d) + 1$
 $\leq c \cdot (\lfloor n/2 \rfloor + \lceil n/2 \rceil) - d - d + 1$
 $\leq cn - d + (1 - d)$
 $\leq cn - d$ falls $d \geq 1$. ✓

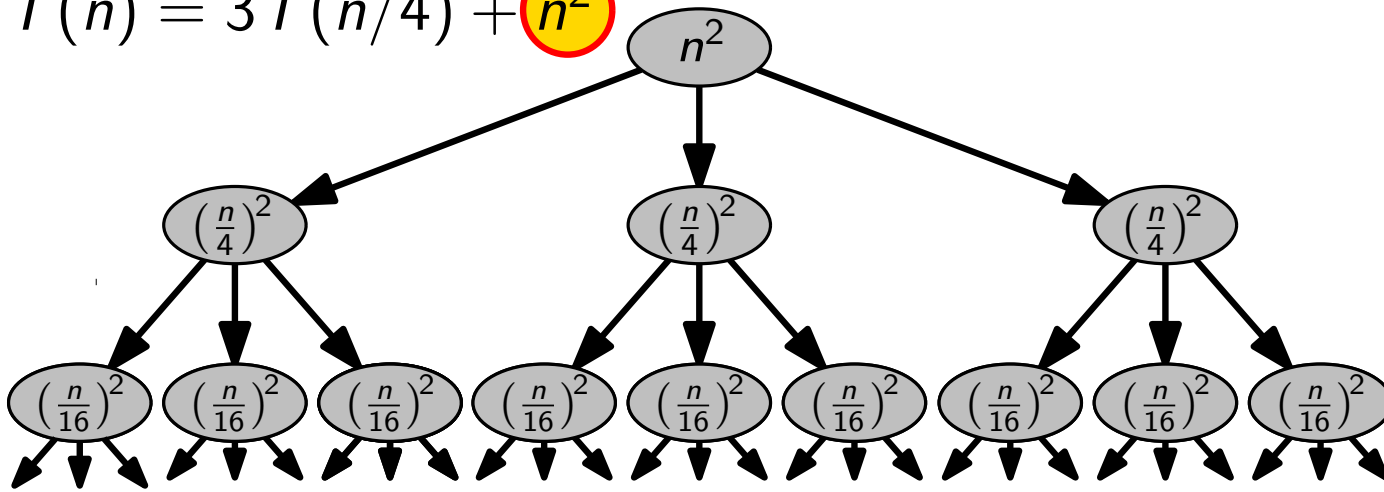
II) Rekursionsbaummethode

Beispiel: $T(n) = 3T(n/4) + n^2$ (*) (mit $T(1) = 1$)



II) Rekursionsbaummethode

$$T(n) = 3T(n/4) + n^2$$



lfd.Nr. Ebene	Anz. Knoten	Beitrag (Ebene)
0	$3^0 = 1$	n^2
1	3^1	$\frac{3}{16} n^2$
2	3^2	$\frac{3^2}{16^2} n^2$
⋮	⋮	⋮
i	3^i	$\frac{3^i}{16^i} n^2$
⋮	⋮	⋮
$\log_4 n$	$3^{\log_4 n}$ $=$ $n^{\log_4 3}$	$n^{\log_4 3}$ <small>vorausgesetzt $T(1) = 1$</small>

0. Summand schon $1n^2!$

unterste Ebene andere Ebenen

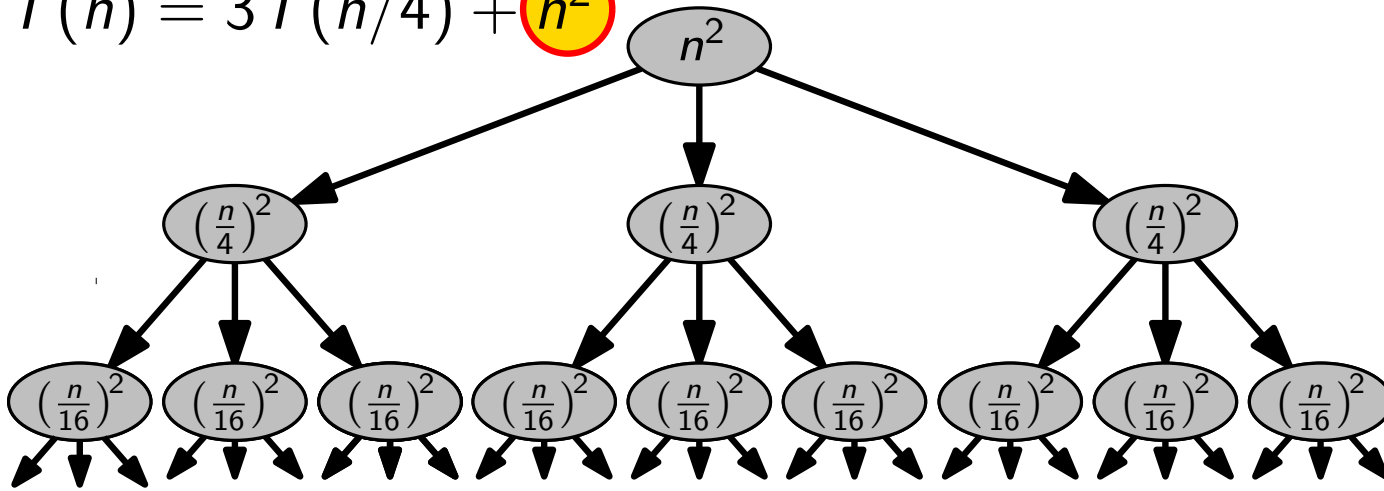
$$\Rightarrow T(n) = n^{\log_4 3} + \sum_{i=0}^{(\log_4 n) - 1} \left(\frac{3}{16}\right)^i n^2 \leq n^{0,793} + n^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i = n^{0,793} + \frac{16}{13} n^2$$

$\Rightarrow T \in \Theta(n^2)!$

geometr. Reihe!!!

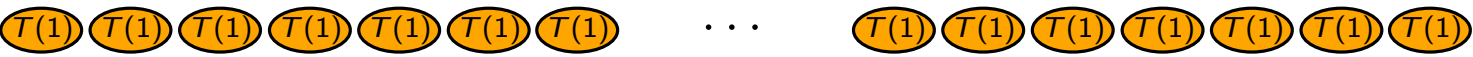
II) Rekursionsbaummethode

$$T(n) = 3T(n/4) + n^2$$



Berechnen Sie mit der Rekursionsbaummethode $T(n) = 2T(n/2) + n \log_2 n$, wobei $T(1) = 0$.

lfd.Nr. Ebene	Anz. Knoten	Beitrag (Ebene)
0	$3^0 = 1$	n^2
1	3^1	$\frac{3}{16} n^2$
2	3^2	$\frac{3^2}{16^2} n^2$
⋮	⋮	⋮
i	3^i	$\frac{3^i}{16^i} n^2$
⋮	⋮	⋮
$\log_4 n$	$3^{\log_4 n} = n^{\log_4 3}$	$n^{\log_4 3}$ <small>vorausgesetzt $T(1) = 1$</small>



0. Summand schon $1n^2!$

unterste Ebene andere Ebenen

$$\Rightarrow T(n) = n^{\log_4 3} + \sum_{i=0}^{(\log_4 n) - 1} \left(\frac{3}{16}\right)^i n^2 \leq n^{0,793} + n^2 \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i = n^{0,793} + \frac{16}{13} n^2 \Rightarrow T \in \Theta(n^2)!$$

geometr. Reihe!!!

III) Meistermethode

Nichts ist praktischer als eine gute Theorie...

Achtung!

Die Methode kann man nur anwenden bei Rekursionen der Art

$$T(n) = aT(n/b) + f(n)$$

wobei $a \geq 1$, $b > 1$ Konst. und $f: \mathbb{N} \rightarrow \mathbb{R}$ *asymptotisch positiv*...

...und auch da nicht in allen Fällen!

III) Meistermethode

Satz: Seien $a \geq 1$, $b > 1$ Konstanten und $f, T: \mathbb{N} \rightarrow \mathbb{R}$ mit

$$T(n) = aT(n/b) + f(n),$$

wobei n/b sowohl für $\lfloor n/b \rfloor$ als auch $\lceil n/b \rceil$ stehen kann.

Dann gilt

$$T \in \begin{cases} \Theta(n^{\log_b a}) & \text{falls } f \in O(n^{(\log_b a) - \varepsilon}) \text{ für ein } \varepsilon > 0. \\ \Theta(n^{\log_b a} \log n) & \text{falls } f \in \Theta(n^{\log_b a}). \\ \Theta(f) & \text{falls } f \in \Omega(n^{(\log_b a) + \varepsilon}) \text{ für ein } \varepsilon > 0 \\ & \text{und die Regularitätsbedingung gilt.} \end{cases}$$

Definition: Die *Regularitätsbedingung* ist erfüllt, falls

$$af(n/b) \leq cf(n)$$

für ein $c < 1$ und für alle großen n .

III) Meistermethode

$$T(n) = aT(n/b) + f(n)$$

$$T \in \begin{cases} \Theta(n^{\log_b a}) & \text{falls } f \in O(n^{(\log_b a) - \varepsilon}) \text{ für ein } \varepsilon > 0. \\ \Theta(n^{\log_b a} \log n) & \text{falls } f \in \Theta(n^{\log_b a}). \\ \Theta(f) & \text{falls } f \in \Omega(n^{(\log_b a) + \varepsilon}) \text{ für ein } \varepsilon > 0 \\ & \text{und die Regularitätsbedingung gilt.} \end{cases}$$

Beispiel: $T(n) = 3T(n/4) + n^2$

$$\Rightarrow a = 3 (\geq 1), b = 4 (> 1) \text{ und } f: n \mapsto n^2.$$

$$\Rightarrow f \in \overset{\Omega}{?}(n^{(\log_4 3) \pm \varepsilon}), \text{ z.B. für } \varepsilon = 1, \text{ da } \log_4 3 < 1.$$

Das ist **Fall 3!** $\Rightarrow T \in \Theta(f) = \Theta(n^2) \quad \square$

Also müssen wir die Regularitätsbedingung testen:

$$3f(n/4) = \frac{3}{16}n^2 \leq c \cdot f(n) = cn^2, \text{ z.B. für } c = \frac{3}{16}.$$

Wichtig: Unser c muss *echt* < 1 sein! ✓

Üben! Hausaufgaben!

Übersicht

- *Substitutionsmethode*

Für „Genies“: Lösung raten, dann per Induktion beweisen!

- *Rekursionsbaummethode*

Etwas umständlich, funktioniert aber immer!

- *Meistermethode*

Funktioniert nur bei Rekursionsgleichungen der Art

$$T(n) = aT(n/b) + f(n) \quad (\text{und auch da nicht immer}).$$

Achtung: Viele verstehen die Bedeutung von ε in den Bedingungen der Fälle 1 & 3 nicht richtig!

Beispiel: $T(n) = 2T(n/2) + n \log_2 n$

$$\Rightarrow n^{\log_b a} = n^{\log_2 2} = n^1$$

$$\text{aber } f(n) = n \log_2 n \notin \Omega(n^{1+\varepsilon}) !!$$

Also können wir die Meistermethode hier **nicht** verwenden!

Grund: $\log n$ wächst langsamer als n^ε , für jedes $\varepsilon > 0$.

PS: Wie könnte man das beweisen?