Datenstrukturen und Algorithmen

Vorlesung 6: Mastertheorem (K4)

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

https://moves.rwth-aachen.de/teaching/ss-18/dsal/

7. Mai 2018

Übersicht

Übersicht

Rekursionsgleichungen

Rekursionsgleichung

Für rekursive Algorithmen wird die Laufzeit meistens durch Rekursionsgleichungen beschrieben.

method foo (int n) {

foo ($\frac{n}{2}$) + - foo ($\frac{n}{2}$) * foo ($\frac{n}{2}$)

}

$$\boxed{T_{foo}(n) = 3 \cdot T_{foo}\left(\frac{n}{2}\right) + \ldots}$$

Rekursionsgleichungen

Rekursionsgleichung

Für rekursive Algorithmen wird die Laufzeit meistens durch Rekursionsgleichungen beschrieben.

Eine Rekursionsgleichung ist eine Gleichung oder eine Ungleichung, die eine Funktion durch ihre eigenen Funktionswerte für kleinere Eingaben beschreibt.

Rekursionsgleichungen

Rekursionsgleichung

Für rekursive Algorithmen wird die Laufzeit meistens durch Rekursionsgleichungen beschrieben.

Eine Rekursionsgleichung ist eine Gleichung oder eine Ungleichung, die eine Funktion durch ihre eigenen Funktionswerte für kleinere Eingaben beschreibt.

Beispiele

- $T(n) = T(\lceil (n-1)/2 \rceil) + 1$ Binäre Suche
- $T(n) = 1\cdot T(n-1) + n - 1$ Bubblesort
- $T(n) = 2 \cdot T(n/2) + n - 1$ Mergesort
- $T(n) = 7 \cdot T(n/2) + c \cdot n^2$ Strassen's Matrixmultiplikation

Die zentrale Frage ist: Wie löst man solche Rekursionsgleichungen?

Die Substitutionsmethode

Substitutionsmethode

Die Substitutionsmethode besteht aus zwei Schritten:

1. Rate die Form der Lösung, durch z. B.:
   - Scharfes Hinsehen, kurze Eingaben ausprobieren und einsetzen
   - Betrachtung des Rekursionsbaums

Die Substitutionsmethode

Substitutionsmethode

Die Substitutionsmethode besteht aus zwei Schritten:

1. Rate die Form der Lösung, durch z. B.:
   - Scharfes Hinsehen, kurze Eingaben ausprobieren und einsetzen
   - Betrachtung des Rekursionsbaums
2. Vollständige Induktion, um die Konstanten zu finden und zu zeigen, dass die Lösung funktioniert.

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \text{ für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.
- Bestimme, ob für ein geeignetes n_0 und für $n \geqslant n_0$ gilt, dass $T(n) \leqslant c \cdot n \cdot \log n$.

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.
- Bestimme, ob für ein geeignetes n_0 und für $n \geqslant n_0$ gilt, dass $T(n) \leqslant c \cdot n \cdot \log n$.
- Stelle fest, dass $T(1) = 1 \leqslant c \cdot 1 \cdot \log 1 = 0$ verletzt ist.

$n_0 \neq 1$

$n_0 > 1$

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.
- Bestimme, ob für ein geeignetes n_0 und für $n \geqslant n_0$ gilt, dass $T(n) \leqslant c \cdot n \cdot \log n$.
- Stelle fest, dass $T(1) = 1 \leqslant c \cdot 1 \cdot \log 1 = 0$ verletzt ist.
- Es gilt: $\underline{T(2)} = \underline{4} \leqslant c \cdot 2 \underbrace{\log 2}_{*}$ und $T(3) = 5 \leqslant \underbrace{c \cdot 3 \log 3}$ für $c \geqslant 2$

$n_0 = 2$

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.
- Bestimme, ob für ein geeignetes n_0 und für $n \geqslant n_0$ gilt, dass $T(n) \leqslant c \cdot n \cdot \log n$.
- Stelle fest, dass $T(1) = 1 \;\leqslant\; c \cdot 1 \cdot \log 1 = 0$ verletzt ist.
- Es gilt: $T(2) = 4 \leqslant c \cdot 2 \log 2$ und $T(3) = 5 \leqslant c \cdot 3 \log 3$ für $c \geqslant 2$
- Überprüfe dann durch Substitution und Induktion (s. nächste Folie)

Die Substitutionsmethode: Beispiel

Beispiel

Betrachte folgende Rekursionsgleichung:

$$T(1) = 1$$
$$T(n) = 2 \cdot T(n/2) + n \quad \text{für } n > 1.$$

- Wir vermuten als Lösung $T(n) \in O(n \cdot \log n)$.
- Dazu müssen wir $T(n) \leqslant c \cdot n \cdot \log n$ zeigen, für ein geeignetes $c > 0$.
- Bestimme, ob für ein geeignetes n_0 und für $n \geqslant n_0$ gilt, dass $T(n) \leqslant c \cdot n \cdot \log n$.
- Stelle fest, dass $T(1) = 1 \;\leqslant\; c \cdot 1 \cdot \log 1 = 0$ verletzt ist.
- Es gilt: $T(2) = 4 \leqslant c \cdot 2 \log 2$ und $T(3) = 5 \leqslant c \cdot 3 \log 3$ für $c \geqslant 2$
- Überprüfe dann durch Substitution und Induktion (s. nächste Folie)
- Damit gilt für jedes $c \geqslant 2$ und $n \geqslant n_0 > 1$, dass $T(n) \leqslant c \cdot n \cdot \log n$.

Die Substitutionsmethode: Beispiel

$T(n) \in O(n \log n)$

Beispiel

$T(n) = 2 \cdot T(n/2) + n$ für $n > 1$, und $T(1) = 1$

$T(\frac{n}{2}) \in O(\frac{n}{2} \log \frac{n}{2})$

$\leq c \cdot \frac{n}{2} \cdot \log \frac{n}{2}$

$$
\begin{aligned}
T(n) &= 2 \cdot T(n/2) + n && \mid \text{ Induktionshypothese} \\
&\leqslant \cancel{2}(c \cdot n/\cancel{2} \cdot \log n/2) + n \\
&= c \cdot n \cdot \log n/2 + n && \mid \text{ log-Rechnung: } (\log \equiv \log_2) \\
& && \mid \ \log n/2 = \log n - \log 2 \\
&= c \cdot n \cdot \log n - c \cdot n \cdot \log 2 + n \\
&\leqslant c \cdot n \cdot \log n - c \cdot n + n && \mid \text{ mit } \boxed{c > 1} \text{ folgt sofort:} \\
&\leqslant c \cdot n \cdot \log n && c = 2
\end{aligned}
$$

Raten der Lösung durch Iteration

Grundidee

Wiederholtes Einsetzen der Rekursionsgleichung in sich selbst, bis man ein Muster erkennt.

$$T(n) = 3 \cdot T\left(\frac{n}{4}\right) + n \qquad \underline{T(1) = c}$$

für n groß genug.

$$(* \quad T\left(\frac{n}{4}\right) = 3 \cdot T\left(\frac{n}{16}\right) + \frac{n}{4} \quad *)$$

$$T(n) = 3 \cdot \left(3 \cdot T\left(\frac{n}{16}\right) + \frac{n}{4}\right) + n$$

$$= 3^2 \, T\left(\frac{n}{16}\right) + \frac{3}{4} \cdot n + n$$

$$= (* \quad T\left(\frac{n}{16}\right) = 3 \cdot T\left(\frac{n}{64}\right) + \frac{n}{16} \quad *)$$

$$= 3^2 \left(3 \, T\left(\frac{n}{64}\right) + \frac{n}{16}\right) + \frac{3}{4} \cdot n + n$$

$$= 3^3 \, T\left(\frac{n}{64}\right) + \left(\frac{3}{4}\right)^2 \cdot n + \left(\frac{3}{4}\right)^1 n + \left(\frac{3}{4}\right)^0 n$$

Nehme mal

$$T(64) = 3^3 \, T(1) + \underbrace{\left(\frac{3}{4}\right)^2 n + \left(\frac{3}{4}\right)^1 n + \left(\frac{3}{4}\right)^0 n}_{\text{wieviele Terme ?}}$$

wie groß

$$\sum_{i=0}^{?} \left(\frac{3}{4}\right)^i n$$

$$T(64) = 3^3 \underbrace{T(1)}_{=c} + \sum_{i=0}^{2} \left(\frac{3}{4}\right)^i \cdot n$$

$$T(64) = \underbrace{3^{\log_4 64}}_{64^{\log_4 3}} = 3^3$$

$$\boxed{T(n) = n^{\log_4 3} + \sum_{i=0}^{?} \left(\frac{3}{4}\right)^i \cdot n}$$

$$? = \log_4 64 = 3$$

$$\longrightarrow \quad ? = \underbrace{\log_4 n - 1}_{\text{Terme } i=0,\ i=1,\ i=2,\ \dots,\ i=\log_4 n - 1}$$

Raten der Lösung durch Iteration

Grundidee

Wiederholtes Einsetzen der Rekursionsgleichung in sich selbst, bis man ein Muster erkennt.

Beispiel

$$\begin{aligned} T(n) &= 3 \cdot T(n/4) + n && \mid \text{Einsetzen} \\ &= 3 \cdot (3 \cdot T(n/16) + n/4)) + n && \mid \text{Nochmal einsetzen} \\ &= 9 \cdot (3 \cdot T(n/64) + n/16)) + 3 \cdot n/4 + n && \mid \text{Vereinfachen} \\ &= 27 \cdot T(n/64) + \left(\frac{3}{4}\right)^2 \cdot n + \left(\frac{3}{4}\right)^1 \cdot n + \left(\frac{3}{4}\right)^0 \cdot n \end{aligned}$$

Raten der Lösung durch Iteration

Grundidee

Wiederholtes Einsetzen der Rekursionsgleichung in sich selbst, bis man ein Muster erkennt.

Beispiel

$$
\begin{aligned}
T(n) &= 3 \cdot T(n/4) + n && \mid \text{Einsetzen} \\
&= 3 \cdot (3 \cdot T(n/16) + n/4)) + n && \mid \text{Nochmal einsetzen} \\
&= 9 \cdot (3 \cdot T(n/64) + n/16)) + 3 \cdot n/4 + n && \mid \text{Vereinfachen} \\
&= 27 \cdot T(n/64) + \left(\frac{3}{4}\right)^2 \cdot n + \left(\frac{3}{4}\right)^1 \cdot n + \left(\frac{3}{4}\right)^0 \cdot n
\end{aligned}
$$

Wir nehmen $T(1) = c$ an und erhalten: $T(n) = \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{4}\right)^i \cdot n + c \cdot n^{\log_4 3}$

Raten der Lösung durch Iteration

Grundidee

Wiederholtes Einsetzen der Rekursionsgleichung in sich selbst, bis man ein Muster erkennt.

Beispiel

$$\begin{aligned}
T(n) &= 3 \cdot T(n/4) + n && \mid \text{Einsetzen} \\
&= 3 \cdot (3 \cdot T(n/16) + n/4)) + n && \mid \text{Nochmal einsetzen} \\
&= 9 \cdot (3 \cdot T(n/64) + n/16)) + 3 \cdot n/4 + n && \mid \text{Vereinfachen} \\
&= 27 \cdot T(n/64) + \left(\frac{3}{4}\right)^2 \cdot n + \left(\frac{3}{4}\right)^1 \cdot n + \left(\frac{3}{4}\right)^0 \cdot n
\end{aligned}$$

Wir nehmen $T(1) = c$ an und erhalten: $T(n) = \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{4}\right)^i \cdot n + c \cdot n^{\log_4 3}$

Diese Aussage kann mit Hilfe der Substitutionsmethode gezeigt werden.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.
   - Die Blätter stellen die Kosten der Basisfälle dar, z. B. $T(0)$ oder $T(1)$.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.
   - Die Blätter stellen die Kosten der Basisfälle dar, z. B. $T(0)$ oder $T(1)$.
2. Wir summieren die Kosten innerhalb jeder Ebene des Baumes.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.
   - Die Blätter stellen die Kosten der Basisfälle dar, z. B. $T(0)$ oder $T(1)$.
2. Wir summieren die Kosten innerhalb jeder Ebene des Baumes.
3. Die Gesamtkosten $:=$ summieren über die Kosten aller Ebenen.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.
   - Die Blätter stellen die Kosten der Basisfälle dar, z. B. $T(0)$ oder $T(1)$.
2. Wir summieren die Kosten innerhalb jeder Ebene des Baumes.
3. Die Gesamtkosten := summieren über die Kosten aller Ebenen.

Wichtiger Hinweis

Ein Rekursionsbaum ist sehr nützlich, um eine Lösung zu raten, die dann mit Hilfe der Substitutionsmethode überprüft werden kann.

Raten der Lösung durch Rekursionsbäume

Grundidee

Stelle das Ineinander-Einsetzen als Baum dar, indem man Buch über das aktuelle Rekursionsargument und die nicht-rekursiven Kosten führt.

Rekursionsbaum

1. Jeder Knoten stellt die Kosten eines Teilproblems dar.
   - Die Wurzel stellt die zu analysierenden Kosten $T(n)$ dar.
   - Die Blätter stellen die Kosten der Basisfälle dar, z. B. $T(0)$ oder $T(1)$.
2. Wir summieren die Kosten innerhalb jeder Ebene des Baumes.
3. Die Gesamtkosten $:=$ summieren über die Kosten aller Ebenen.

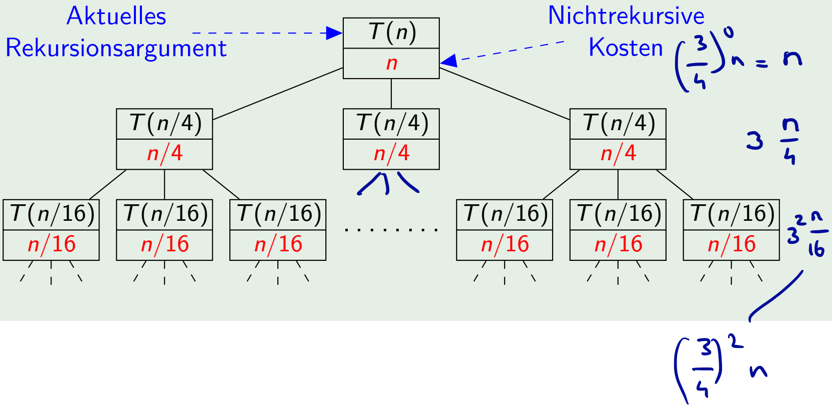
Wichtiger Hinweis

Ein Rekursionsbaum ist sehr nützlich, um eine Lösung zu raten, die dann mit Hilfe der Substitutionsmethode überprüft werden kann.
Der Baum selber reicht jedoch meistens nicht als Beweis.

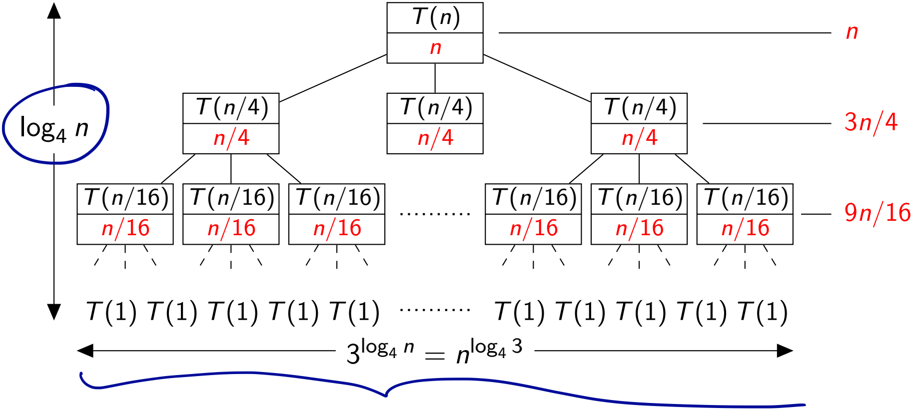
Rekursionsbaum: Beispiel

Beispiel

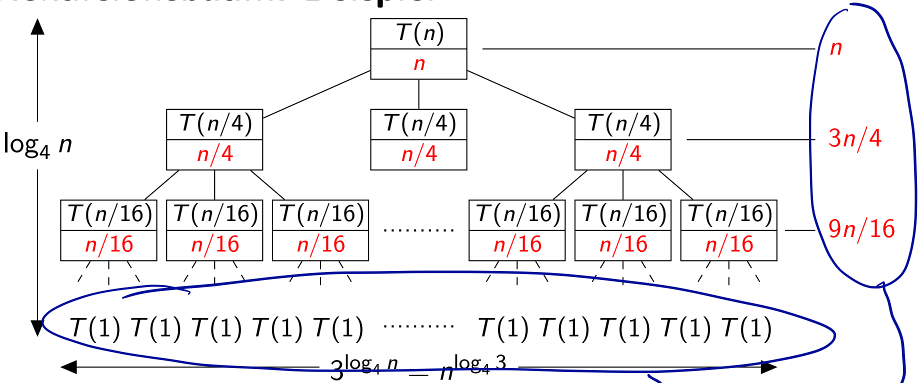
Der Rekursionsbaum von $T(n) = 3 \cdot T(n/4) + n$ sieht etwa so aus:



Rekursionsbaum: Beispiel



Rekursionsbaum: Beispiel



$$T(n) = \underbrace{\sum_{i=0}^{\log_4 n - 1}}_{\substack{\text{Summe über} \\ \text{alle Ebenen}}} \underbrace{\left(\frac{3}{4}\right)^i \cdot n}_{\substack{\text{Kosten pro} \\ \text{Ebene}}} + \underbrace{c \cdot n^{\log_4 3}}_{\substack{\text{Gesamtkosten} \\ \text{für die Blätter} \\ \text{mit } T(1) = c}}$$

Rekursionsbaum: Beispiel

$\boxed{T(n) \in O(n)}$

Eine obere Schranke für die Komplexität erhält man nun folgendermaßen:

$$
\begin{aligned}
T(n) &= \boxed{\sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{4}\right)^i \cdot n + c \cdot n^{\log_4 3}} && \mid \text{ Vernachlässigen kleinerer Terme} \\
&< \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i \cdot n + c \cdot n^{\log_4 3} && \mid \text{ Geometrische Reihe} \\
&< \frac{1}{\underline{1-(3/4)}} \cdot n + c \cdot n^{\log_4 3} && \mid \text{ Umformen} \\
&\le \boxed{4 \cdot n} + \boxed{c \cdot n^{\log_4 3}} && \mid \text{ Asymptotische Ordnung bestimmen, setze ein, dass } \log_4 3 < 1
\end{aligned}
$$

< 1

$=$

$3^{\log_4 n}$

$|a| < 1$ dann $\sum_{i=0}^{\infty} d \cdot a^i = \frac{d}{1-a}$

$T(n) \in O(n)$.

$\log_4 3 < 1$ → Koste der Blätter

Koste Ebenen

Korrektheit

Wir können die Substitutionsmethode benutzen, um die Vermutung zu bestätigen, dass:

$T(n) \in O(n)$ eine obere Schranke von $T(n) = 3 \cdot T(n/4) + n$ ist.

Korrektheit

Wir können die Substitutionsmethode benutzen, um die Vermutung zu bestätigen, dass:

$T(n) \in O(n)$ eine obere Schranke von $T(n) = 3 \cdot T(n/4) + n$ ist.

$$T(n) = 3 \cdot \underline{T(n/4)} + n \qquad | \text{ Induktionshypothese}$$

$$\leqslant 3\underline{d \cdot n/4} + n$$

$$= \frac{3}{4} d \cdot n + n$$

$$= \left(\frac{3}{4} d + 1\right) \cdot n \qquad | \text{ mit } d \geqslant 4 \text{ folgt sofort:}$$

$$\leqslant d \cdot n$$

$T(n) \in O(n)$

d.h $T(\frac{n}{4}) \in O(\frac{n}{4})$

d.h $\boxed{T(\frac{n}{4}) \leq d \cdot \frac{n}{4}}$ für eine konstante $d > 0$

$\frac{3}{4} d + 1 \leq d$

$\Leftrightarrow 1 \leq \frac{d}{4} \Leftrightarrow d \geq 4$

Korrektheit

Wir können die Substitutionsmethode benutzen, um die Vermutung zu bestätigen, dass:

$T(n) \in O(n)$ eine obere Schranke von $T(n) = 3 \cdot T(n/4) + n$ ist.

$$\begin{aligned} T(n) &= 3 \cdot T(n/4) + n && \mid \text{Induktionshypothese} \\ &\leqslant 3d \cdot n/4 + n \\ &= \frac{3}{4} d \cdot n + n \\ &= \left(\frac{3}{4} d + 1\right) \cdot n && \mid \text{mit } d \geqslant 4 \text{ folgt sofort:} \\ &\leqslant d \cdot n \end{aligned}$$

Und wir stellen fest, dass es ein n_0 gibt, sodass $T(n_0) \leqslant d \cdot n_0$ ist für $d \geqslant 4$.

Mastertheorem

Allgemeines Format der Rekursionsgleichung

Eine Rekursionsgleichung für die Komplexitätsanalyse sieht meistens folgendermaßen aus:

$$T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)$$

wobei $b > 0$, $c > 1$ gilt und $f(n)$ eine gegebene Funktion ist.

Anzahl der Teilprobleme

Eingabegröße der rekursiven Aufrufe

nicht-rekursive Kosten

Mastertheorem

Allgemeines Format der Rekursionsgleichung

Eine Rekursionsgleichung für die Komplexitätsanalyse sieht meistens folgendermaßen aus:

$$T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)$$

wobei $b > 0$, $c > 1$ gilt und $f(n)$ eine gegebene Funktion ist.

Intuition:

- Das zu analysierende Problem teilt sich jeweils in b Teilprobleme auf.

Mastertheorem

Allgemeines Format der Rekursionsgleichung

Eine Rekursionsgleichung für die Komplexitätsanalyse sieht meistens folgendermaßen aus:

$$T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)$$

wobei $b > 0$, $c > 1$ gilt und $f(n)$ eine gegebene Funktion ist.

Intuition:

- Das zu analysierende Problem teilt sich jeweils in b Teilprobleme auf.
- Jedes dieser Teilprobleme hat die Größe $\frac{n}{c}$.

Mastertheorem

Allgemeines Format der Rekursionsgleichung

Eine Rekursionsgleichung für die Komplexitätsanalyse sieht meistens folgendermaßen aus:

$$T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)$$

wobei $b > 0$, $c > 1$ gilt und $f(n)$ eine gegebene Funktion ist.

Intuition:

- Das zu analysierende Problem teilt sich jeweils in b Teilprobleme auf.
- Jedes dieser Teilprobleme hat die Größe $\frac{n}{c}$.
- Die Kosten für das Aufteilen eines Problems und Kombinieren der Teillösungen sind $f(n)$.

Das Mastertheorem

$$\boxed{T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)} \text{ mit } b \geqslant 1 \text{ und } c > 1.$$

- Anzahl der Blätter im Rekursionsbaum: n^E mit $E = \log_2 b / \log_2 c$.

$n^{\log_4 3}$

$\log_4 3 = \frac{\log_2 3}{\log_2 4}$

$T(n) = 3 \cdot T\left(\frac{n}{4}\right)$

Das Mastertheorem

Summe über alle Ebenen

$$\boxed{T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)} \text{ mit } b \geqslant 1 \text{ und } c > 1.$$

- Anzahl der Blätter im Rekursionsbaum: n^E mit $E = \log b / \log c$.

Mastertheorem

	Wenn	Dann
1.	$f(n) \in O(n^{E-\varepsilon})$ für ein $\varepsilon > 0$	$T(n) \in \Theta(n^E)$

Das Mastertheorem

$$\boxed{T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)} \text{ mit } b \geqslant 1 \text{ und } c > 1.$$

- Anzahl der Blätter im Rekursionsbaum: n^E mit $E = \log b / \log c$.

Mastertheorem

	Wenn	Dann
1.	$f(n) \in O(n^{E-\varepsilon})$ für ein $\varepsilon > 0$	$T(n) \in \Theta(n^E)$
2.	$\underline{f(n)} \in \underline{\Theta(n^E)}$	$T(n) \in \underline{\Theta(n^E \cdot \log n)}$

Das Mastertheorem

$$\boxed{T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)} \text{ mit } b \geqslant 1 \text{ und } c > 1.$$

- Anzahl der Blätter im Rekursionsbaum: n^E mit $E = \log b / \log c$.

Mastertheorem

	Wenn	Dann
	f(n) ist strikt langsamer	
1.	$f(n) \in O(n^{E-\varepsilon})$ für ein $\varepsilon > 0$	$T(n) \in \Theta(n^E)$
2.	$f(n) \in \Theta(n^E)$ „genau so schnell"	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n) \in \Omega(n^{E+\varepsilon})$ für ein $\varepsilon > 0$ und $b \cdot f(n/c) \leqslant d \cdot f(n)$ für ein $d < 1$ und n hinreichend groß	$T(n) \in \Theta(f(n))$

f(n) wächst wirklich schneller als n^E

Das Mastertheorem

$$\boxed{T(n) = b \cdot T\left(\frac{n}{c}\right) + f(n)} \text{ mit } b \geqslant 1 \text{ und } c > 1.$$

- Anzahl der Blätter im Rekursionsbaum: n^E mit $E = \log b / \log c$.

Mastertheorem

	Wenn	Dann
1.	$f(n) \in O(n^{E-\varepsilon})$ für ein $\varepsilon > 0$	$T(n) \in \Theta(n^E)$
2.	$f(n) \in \Theta(n^E)$	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n) \in \Omega(n^{E+\varepsilon})$ für ein $\varepsilon > 0$ und $b \cdot f(n/c) \leqslant d \cdot f(n)$ für ein $d < 1$ und n hinreichend groß	$T(n) \in \Theta(f(n))$

- Bemerke, dass das Mastertheorem nicht alle Fälle abdeckt.

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

nicht-rekursive Kosten

Anzahl der Blätter

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ polynomial kleiner ist als n^E	$T(n) \in \Theta(n^E)$

$f(n) \in O(n^{E-\varepsilon})$

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ polynomial kleiner ist als n^E	$T(n) \in \Theta(n^E)$
2.	$f(n)$ und n^E die "gleiche Größe" haben	$T(n) \in \Theta(n^E \cdot \log n)$

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ polynomial kleiner ist als n^E	$T(n) \in \Theta(n^E)$
2.	$f(n)$ und n^E die gleiche Größe haben	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n)$ ist polynomial größer als n^E und erfüllt $b \cdot f(n/c) \leqslant d \cdot f(n)$ $d < 1$	$T(n) \in \Theta(f(n))$

$\exists \varepsilon > 0. \quad f(n) \in \Omega(n^{E+\varepsilon})$

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ ~~polynomial~~ kleiner ist als n^E	$T(n) \in \Theta(n^E)$
2.	$f(n)$ und n^E die gleiche Größe haben	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n)$ ist polynomial größer als n^E und erfüllt $b \cdot f(n/c) \leqslant d \cdot f(n)$	$T(n) \in \Theta(f(n))$

Nicht abgedeckte Fälle:

1. $f(n)$ ist kleiner als n^E, jedoch nicht polynomiell kleiner.

$\cancel{f(n) \in \Theta(n^{E-\varepsilon})}$ für $\varepsilon > 0$

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ polynomial kleiner ist als n^E	$T(n) \in \Theta(n^E)$
2.	$f(n)$ und n^E die gleiche Größe haben	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n)$ ist polynomial größer als n^E und erfüllt $b \cdot f(n/c) \leqslant d \cdot f(n)$	$T(n) \in \Theta(f(n))$

Nicht abgedeckte Fälle:

1. $f(n)$ ist kleiner als n^E, jedoch nicht polynomiell kleiner.
2. $f(n)$ ist größer als n^E, jedoch nicht polynomiell größer.

$f(n) \in \cancel{\Omega(n^{E+\varepsilon})}$ für $\varepsilon > 0$

Das Mastertheorem verstehen

In jedem der 3 Fälle wird die Funktion $f(n)$ mit $n^E = n^{\log_c b}$ verglichen.

Mastertheorem: Intuition

	Wenn	Dann
1.	$f(n)$ polynomial kleiner ist als n^E	$T(n) \in \Theta(n^E)$
2.	$f(n)$ und n^E die gleiche Größe haben	$T(n) \in \Theta(n^E \cdot \log n)$
3.	$f(n)$ ist polynomial größer als n^E und erfüllt $b \cdot f(n/c) \leqslant d \cdot f(n)$ für $d < 1$	$T(n) \in \Theta(f(n))$

Nicht abgedeckte Fälle:

1. $f(n)$ ist kleiner als n^E, jedoch nicht polynomiell kleiner.
2. $f(n)$ ist größer als n^E, jedoch nicht polynomiell größer.
3. $f(n)$ ist polynomiell größer als n^E, erfüllt nicht $b \cdot f(n/c) \leqslant d \cdot f(n)$.

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $\underbrace{f(n) = n}$; $E = \log \underset{b}{4} / \log \underset{c}{2} = 2$.

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $f(n) = n$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n \in O(n^{2-\varepsilon})$, gilt Fall 1: $\boxed{T(n) \in \Theta(n^2)}$

wenn $\underbrace{f(n) \in O(n^{2-\varepsilon})}$, dann $T(n) \in \Theta(n^E)$

$E = 2$

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $f(n) = n$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n \in O(n^{2-\varepsilon})$, gilt Fall 1: $\boxed{T(n) \in \Theta(n^2)}$

Beispiel

$T(n) = 4 \cdot T(n/2) + n^2$

$\uparrow$ b $\quad$ $\uparrow$ c $\quad$ Ξ

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $f(n) = n$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n \in O(n^{2-\varepsilon})$, gilt Fall 1: $\boxed{T(n) \in \Theta(n^2)}$

Beispiel

$T(n) = 4 \cdot T(n/2) + n^2$

- Somit: $b = 4$, $c = 2$ und $f(n) = n^2$; $E = \log 4 / \log 2 = 2$.

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $f(n) = n$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n \in O(n^{2-\varepsilon})$, gilt Fall 1: $\boxed{T(n) \in \Theta(n^2)}$

Beispiel

$T(n) = 4 \cdot T(n/2) + n^2$

- Somit: $b = 4$, $c = 2$ und $f(n) = n^2$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n^2 \notin O(n^{2-\varepsilon})$, gilt Fall 1 nicht.

Anwendung des Mastertheorems

Beispiel

$T(n) = 4 \cdot T(n/2) + n$

- Somit: $b = 4$, $c = 2$ und $f(n) = n$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n \in O(n^{2-\varepsilon})$, gilt Fall 1: $\boxed{T(n) \in \Theta(n^2)}$

Beispiel

$T(n) = 4 \cdot T(n/2) + n^2$

- Somit: $b = 4$, $c = 2$ und $f(n) = n^2$; $E = \log 4 / \log 2 = 2$.
- Da $f(n) = n^2 \notin O(n^{2-\varepsilon})$, gilt Fall 1 nicht.
- Aber weil $\underbrace{f(n) = n^2}$ $\in \underbrace{\Theta(n^2)}$, gilt Fall 2: $\boxed{T(n) \in \Theta(n^2 \cdot \log n)}$

$n^E \cdot \log n$

$\Theta(n^E) = \Theta(n^2)$

Anwendung des Mastertheorems

Beispiel

$T(n) = \underline{4 \cdot T(n/2)} + \underline{n^3}$

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.
- Wegen $E = 2$ gelten Fälle 1 und 2 offenbar nicht.

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.
- Wegen $E = 2$ gelten Fälle 1 und 2 offenbar nicht.
- Da $f(n) = \underline{n^3} \in \underline{\Omega(n^{2+\varepsilon})}$ für $\varepsilon = 1$, könnte Fall 3 gelten.

Wenn $f(n) \in \Omega(n^{E+\varepsilon})$ für $\varepsilon > 0$ ✓

<u>und</u> $b \cdot f(\frac{n}{c}) \leq d \cdot f(n)$ für $d < 1$ 2

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.
- Wegen $E = 2$ gelten Fälle 1 und 2 offenbar nicht.
- Da $f(n) = n^3 \in \Omega(n^{2+\varepsilon})$ für $\varepsilon = 1$, könnte Fall 3 gelten.
- Überprüfe: gilt $f(n/2) \leqslant \frac{d}{4} \cdot f(n)$ für ein $d < 1$ und hinreichend grosse n?

$$b \cdot f\left(\frac{n}{c}\right) \leq d \cdot f(n)$$

$$\Leftrightarrow \quad 4 \cdot f\left(\frac{n}{2}\right) \leq d \cdot f(n)$$

$$\Leftrightarrow \quad f\left(\frac{n}{2}\right) \leq \frac{d}{4} \cdot f(n)$$

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.
- Wegen $E = 2$ gelten Fälle 1 und 2 offenbar nicht.
- Da $f(n) = n^3 \in \Omega(n^{2+\varepsilon})$ für $\varepsilon = 1$, könnte Fall 3 gelten.
- Überprüfe: gilt $f(n/2) \leqslant \frac{d}{4} \cdot f(n)$ für ein $d < 1$ und hinreichend grosse n?
- Dies liefert $\frac{1}{8} n^3 \leqslant \frac{d}{4} \cdot n^3$, und dies gilt für alle $\frac{1}{2} \leqslant d < 1$ (und n)

$f(\frac{n}{2}) = (\frac{n}{2})^3 = \frac{1}{8} n^3$

Anwendung des Mastertheorems

Beispiel

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

- Somit: $b = 4$, $c = 2$ und $f(n) = n^3$; $E = \log 4 / \log 2 = 2$.
- Wegen $E = 2$ gelten Fälle 1 und 2 offenbar nicht.
- Da $f(n) = n^3 \in \Omega(n^{2+\varepsilon})$ für $\varepsilon = 1$, könnte Fall 3 gelten.
- Überprüfe: gilt $f(n/2) \leqslant \frac{d}{4} \cdot f(n)$ für ein $d < 1$ und hinreichend grosse n?
- Dies liefert $\frac{1}{8} n^3 \leqslant \frac{d}{4} \cdot n^3$, und dies gilt für alle $\frac{1}{2} \leqslant d < 1$ (und n)
- Somit gilt Fall 3 tatsächlich und wir folgern: $\boxed{T(n) \in \Theta(n^3)}$

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$$T(n) = \underset{b}{4} \cdot T\left(n/\underset{c}{2}\right) + \frac{n^2}{\log n}$$

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log n}$

- Also gilt: $b = 4$, $c = 2$ und $f(n) = n^2/\log n$; $E = 2$.

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log n}$

- Also gilt: $b = 4$, $c = 2$ und $f(n) = n^2/\log n$; $E = 2$.

Fall 1 ist nicht anwendbar:

$$n^2/\log n \notin O(n^{2-\varepsilon}), \text{ da } \underline{f(n)/n^2} = \underline{(\log n)^{-1}} \notin \underline{O(n^{-\varepsilon})}.$$

$f(n) \in O(n^{2-\varepsilon})$

$\frac{n^2}{\log n} \in O(n^{2-\varepsilon})$

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log n}$

- Also gilt: $b = 4$, $c = 2$ und $f(n) = n^2/\log n$; $E = 2$.

Fall 1 ist nicht anwendbar:

$$n^2/\log n \notin O(n^{2-\varepsilon}), \text{ da } f(n)/n^2 = (\log n)^{-1} \notin O(n^{-\varepsilon}).$$

Fall 2 ist nicht anwendbar: $\underbrace{n^2/\log n}_{f(n)} \notin \Theta(\underset{\substack{\uparrow \\ n^E}}{n^2})$.

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log n}$

- Also gilt: $b = 4$, $c = 2$ und $f(n) = n^2 / \log n$; $E = 2$.

Fall 1 ist nicht anwendbar:

$$n^2 / \log n \notin O(n^{2-\varepsilon}), \text{ da } f(n)/n^2 = (\log n)^{-1} \notin O(n^{-\varepsilon}).$$

Fall 2 ist nicht anwendbar: $n^2 / \log n \notin \Theta(n^2)$.

Fall 3 ist nicht anwendbar:

1 $$f(n) \notin \Omega(\underbrace{n^{2+\varepsilon}}_{\varepsilon > 0}), \text{ da } \underbrace{f(n)/n^2} = \underbrace{(\log n)^{-1}} \notin O(n^{+\varepsilon}).$$

Das Mastertheorem ist nicht immer anwendbar

Beispiel

$T(n) = 4 \cdot T(n/2) + \frac{n^2}{\log n}$

- Also gilt: $b = 4$, $c = 2$ und $f(n) = n^2 / \log n$; $E = 2$.

Fall 1 ist nicht anwendbar:

$$n^2 / \log n \notin O(n^{2-\varepsilon}), \text{ da } f(n)/n^2 = (\log n)^{-1} \notin O(n^{-\varepsilon}).$$

Fall 2 ist nicht anwendbar: $n^2 / \log n \notin \Theta(n^2)$.

Fall 3 ist nicht anwendbar:

$$f(n) \notin \Omega(n^{2+\varepsilon}), \text{ da } f(n)/n^2 = (\log n)^{-1} \notin O(n^{+\varepsilon}).$$

$\Longrightarrow$ Das Mastertheorem hilft hier überhaupt nicht weiter!

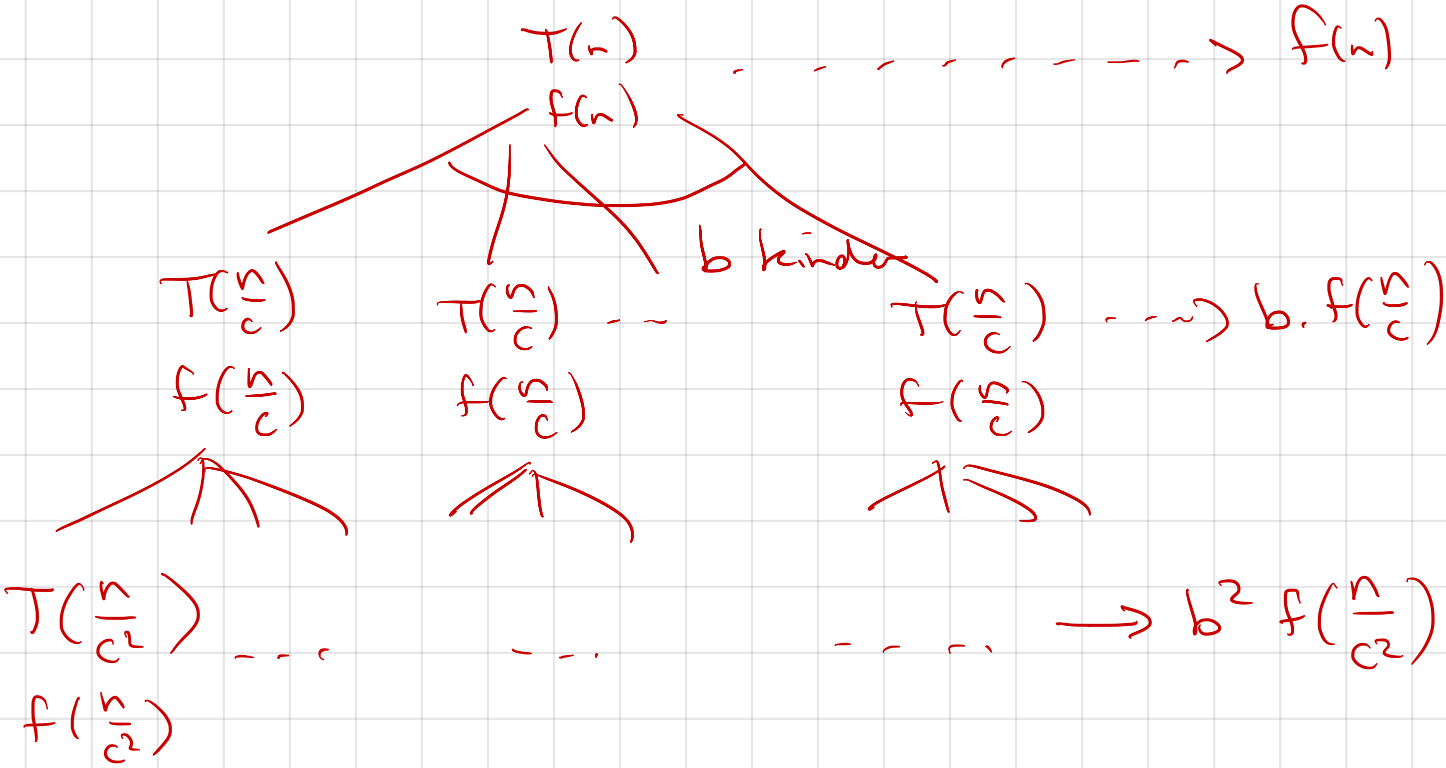
- Durch Substitution erhält man: $\boxed{T(n) \in \Theta(n^2 \cdot \log \log n)}$

Mastertheorem: Beweis

Sei $f(n)$ eine nicht-negative Funktion, die definiert ist über den Potenzen von c $(c^0, c^1, c^2, \ldots)$. Definiere $T(n)$ über die Potenze von c wie folgt:

$$T(n) = \begin{cases} k & \text{falls } n = 1 = c^0 \\ b \cdot T\left(\frac{n}{c}\right) + f(n), & \text{falls } n = c^i \\ & i \in \mathbb{N}_{>0} \end{cases}$$

Dann: $$T(n) = \underbrace{\sum_{j=0}^{\log_c n - 1}}_{\text{Summe über alle Ebenen}} \underbrace{b^j \cdot f\left(\frac{n}{c^j}\right)}_{\text{Kosten pro Ebene}} + \underbrace{k \cdot n^{\log_c b}}_{\text{Gesamtkosten der Blätter}}$$



Fall 1 nehme an, dass $f(n) \in O(n^{E-\varepsilon})$ (*) für $\varepsilon > 0$

Zu zeigen: $\boxed{T(n) \in \Theta(n^{E})}$

Beweis Sei $g(n) = \sum_{j=0}^{\log_c n-1} b^j \cdot \underbrace{f\left(\frac{n}{c^j}\right)}_{=}$ (**)

$f(n) \in O(n^{E-\varepsilon}) = O(n^{\log_c b - \varepsilon})$

dann $f\left(\frac{n}{c^j}\right) \in O\left(\left(\frac{n}{c^j}\right)^{\log_c b-\varepsilon}\right)$ (***)

(**) + (***) führt zu:

$g(n) = O\left(\underbrace{\sum_{j=0}^{\log_c n-1} b^j \cdot \left(\frac{n}{c^j}\right)^{\log_c b-\varepsilon}}_{}\right)$

können wir das vereinfachen?

$$\sum_{j=0}^{\log_c n-1} b^j \left(\frac{n}{c^j}\right)^{\log_c b-\varepsilon}$$

$$= n^{\log_c b-\varepsilon} \cdot \sum_{j=0}^{\log_c n-1} b^j \left(\frac{1}{c^j}\right)^{\log_c b-\varepsilon}$$

$$= n^{\log_c b-\varepsilon} \sum_{j=0}^{\log_c n-1} \cancel{b^j} \cdot \left(\frac{c^\varepsilon}{\cancel{c^{\log_c b}}}\right)^j$$

$$c^{\log_c b} = b$$

$$= n^{\log_c b-\varepsilon} \sum_{j=0}^{\log_c n-1} (c^\varepsilon)^j$$

= (* geometrische Reihe *)

$$= n^{\log_c b-\varepsilon} \cdot \left(\frac{c^{\varepsilon \log_c n} - 1}{c^\varepsilon - 1}\right)$$

$$= \underbrace{n^{\log_c b-\varepsilon}} \cdot \left(\frac{n^\varepsilon - 1}{c^\varepsilon - 1}\right) \qquad \varepsilon > 0$$

$$= \frac{n^{\log_c b} - n^{\log_c b-\varepsilon}}{c^\varepsilon - 1} \in O\left(n^{\log_c b}\right)$$

$$\underbrace{T(n) = \underbrace{\sum_{j=0}^{\log_c n - 1} b^j \, f\left(\frac{n}{c^j}\right)}_{\substack{g(n) \\ \in O\left(n^{\log_c b}\right) \\ = O\left(n^E\right)}} + \underbrace{k \, n^{\log_c b}}_{\Theta\left(n^E\right)}}$$

$$T(n) \in \Theta\left(n^{\log_c b}\right) \qquad \boxtimes$$

Nächste Vorlesung

Nächste Vorlesung

Freitag 11. Mai, 13:15 (Hörsaal H01). Bis dann!