Lösen elementarer Rekursionen

Inhomogene lineare Rekursionen mit konstanten Koeffizienten

Satz

Sei $(a_n^{(p)})_{n\geq 0}$ *Lösung der inhomogenen linearen Rekursion*

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = s_n, \quad n \geq 0,$$

mit $c_0, c_1, \ldots, c_k \in \mathbb{R}$*, wobei* $c_0 \neq 0$ *und* $c_k \neq 0$*, und* L *die Lösungsmenge der Rekursion*

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0, \quad n \geq 0.$$

Dann ist

$$(a_n^{(p)})_{n\geq 0} + L$$

die Lösungsmenge der inhomogenen Rekursion.

Lösen elementarer Rekursionen

Beweis: Sei $(a_n)_{n\geq 0} \in M$. Dann gilt

$$\begin{array}{rcccrl} a_{n+k}c_k & + & \cdots & + & a_n c_0 & = s_n \\ a^{(p)}_{n+k}c_k & + & \cdots & + & a^{(p)}_n c_0 & = s_n \\ \hline (a_{n+k} - a^{(p)}_{n+k})c_k & + & \cdots & + & (a_n - a^{(p)}_n)c_0 & = 0 \end{array}$$

Daraus folgt $(a_n - a^{(p)}_n)_{n\geq 0} \in L$ und somit $(a_n)_{n\geq 0} \in (a^{(p)}_n)_{n\geq 0} + L$.

Sei nun $(a_n)_{n\geq 0} \in (a^{(p)}_n)_{n\geq 0} + L$.
Dann ist $a_n = a^{(p)}_n + b_n$ mit $(b_n)_{n\geq 0} \in L$ und es gilt

$$\begin{array}{rcccrl} a^{(p)}_{n+k}c_k & + & \cdots & + & a^{(p)}_n c_0 & = s_n \\ b_{n+k}c_k & + & \cdots & + & b_n c_0 & = 0 \\ \hline a_{n+k}c_k & + & \cdots & + & a_n c_0 & = s_n \end{array}$$

Daraus folgt $(a_n)_{n\geq 0} \in M$. $\square$

Spezielle Störfunktionen – Ansatzmethode

Satz

Gesucht ist eine partikuläre Lösung von

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = s_n, \quad n \geq 0,$$

wobei $s_n = P_j(n)\alpha^n$ ($P_j(x)$ sei ein Polynom vom Grad j).
Dann führt folgender Ansatz zum Ziel:

$$a_n^{(p)} = q_j(n)\alpha^n n^{\nu(\alpha)},$$

wobei $q_j(x)$ ein Polynom vom Grad j mit unbestimmten Koeffizienten und α eine Nullstelle der Vielfachheit $\nu(\alpha)$ von $\chi(\lambda)$ ist.

Das Superpositionsprinzip

Satz

Wenn $(\alpha_n)_{n\geq 0}$ *eine partikuläre Lösung von*

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = s_n, \quad n \geq 0,$$

ist und $(\beta_n)_{n\geq 0}$ *eine partikuläre Lösung von*

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = t_n, \quad n \geq 0,$$

dann ist $(\alpha_n + \beta_n)_{n\geq 0}$ *eine partikuläre Lösung von*

$$c_k a_{n+k} + c_{k-1} a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = s_n + t_n, \quad n \geq 0.$$

Lösen elementarer Rekursionen

Beweis: Für $(\alpha_n)_{n\geq 0}$ bzw. $(\beta_n)_{n\geq 0}$ gilt

$$\begin{array}{rcccrcl}
\alpha_{n+k}c_k & + & \cdots & + & \alpha_n c_0 & = & s_n \\
\beta_{n+k}c_k & + & \cdots & + & \beta_n c_0 & = & t_n \\
\hline
(\alpha_{n+k}+\beta_{n+k})c_k & + & \cdots & + & (\alpha_n+\beta_n)c_0 & = & s_n+t_n
\end{array}$$

$\square$

Beispiel

$a_n - a_{n-1} - 8a_{n-2} + 12a_{n-3} = n4^n + 2^n$

$\chi(\lambda) = \lambda^3 - \lambda^2 - 8\lambda + 12 = (\lambda-2)^2(\lambda+3)$

Lösung der homogenen Rekursion $a_n - a_{n-1} - 8a_{n-2} + 12a_{n-3} = 0$:
$a_n^{(h)} = K_1 \cdot 2^n + K_2 \cdot n2^n + K_3 \cdot (-3)^n$

Ansatz für $n4^n$: $a_n^{(1)} = (An+B)4^n$

Ansatz für 2^n: $a_n^{(2)} = n^2 2^n$

Lösung: $a_n = K_1 \cdot 2^n + K_2 \cdot n2^n + K_3 \cdot (-3)^n + a_n^{(1)} + a_n^{(2)}$

Lösen elementarer Rekursionen

Lineare Rekursionen erster Ordnung

homogen: $a_{n+1} = \alpha_n a_n,\ n \geq 0$

Lösung durch Iteration: $a_n = C \prod\limits_{j=0}^{n-1} \alpha_j$

inhomogen: $a_{n+1} = \alpha_n a_n + \beta_n,\ n \geq 0.$

Lösungsstruktur:

$$a_n = a_n^{(h)} + a_n^{(p)}$$

Finden einer partikulären Lösung durch Variation der Konstanten:
Ansatz: $a_n^{(p)} = C_n \prod_{j=0}^{n-1} \alpha_j$

Lösen elementarer Rekursionen

$a_{n+1} = \alpha_n a_n + \beta_n,\ n \geq 0.$
Ansatz: $a_n^{(p)} = C_n \prod_{j=0}^{n-1} \alpha_j$

$$C_{n+1} \prod_{j=0}^{n} \alpha_j = \alpha_n C_n \prod_{j=0}^{n-1} \alpha_j + \beta_n$$

$$C_{n+1} = C_n + \frac{\beta_n}{\prod_{j=0}^{n} \alpha_j}$$

$$C_n = \sum_{\ell=0}^{n-1} \frac{\beta_\ell}{\prod_{j=0}^{\ell} \alpha_j} + C_0$$

Allgemeine Lösung:

$$a_n = \left(\prod_{j=0}^{n-1} \alpha_j \right) \cdot \sum_{\ell=0}^{n-1} \frac{\beta_\ell}{\prod\limits_{j=0}^{\ell} \alpha_j} + C \prod_{j=0}^{n-1} \alpha_j$$

Kombinatorische Grundprobleme

Grundlegende Abzählregeln

- Summenregel: Falls $A \cap B = \emptyset$, dann gilt $|A \cup B| = |A| + |B|$
- Produktregel: $|A \times B| = |A| \cdot |B|$
- Gleichheitsregel: $|A| = |B|$ genau dann, wenn es eine Bijektion $f : A \to B$ gibt.

Notationen:

- $n! := n(n-1)!, \ n \geq 1, \qquad 0! := 1$
- $\binom{n}{k} := \dfrac{n!}{k!(n-k)!}$

Anordnungs- und Auswahlprobleme

Sei $A = \{a_1, a_2, \ldots, a_n\}$.

- Anordnungen ohne Einschränkung (Variationen mit Wh.):
  $(a_{i_1}, \ldots, a_{i_k})$; Anzahl: $|A^k| = n^k$.
- Anordnungen verschiedener Elemente (Variationen ohne Wh.):
  $(a_{i_1}, \ldots, a_{i_k})$ mit paarweise verschiedenen Indizes;
  Anzahl $= n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$.
- Permutationen von A:

$$\begin{pmatrix} a_1 & \ldots & a_n \\ \pi(a_1) & \ldots & \pi(a_n) \end{pmatrix};$$

Anzahl $= n!$.

Kombinatorische Grundprobleme

- Permutationen einer Multimenge: $\{a_1^{k_1}, a_2^{k_2}, \ldots, a_n^{k_n}\}$

$$\text{Anzahl} = \frac{(k_1 + k_2 + \cdots + k_n)!}{k_1!k_2! \cdots k_n!}$$

- Auswahlen einer Teilmenge (Kombinationen ohne Wh.):
  $\{a_{i_1}, \cdots, a_{i_k}\} \subseteq A$; Anzahl $= \binom{n}{k}$.
- Auswahlen einer Teilmultimenge (Kombinationen mit Wh.):
  $\{a_1^{k_1}, a_2^{k_2}, \ldots, a_n^{k_n}\}$ mit $k_1 + k_2 + \cdots + k_n = k$;

  Anzahl$=\binom{n+k-1}{n-1} = \binom{n+k-1}{k}$.

Weitere Zählmethoden

•) Doppeltes Zählen

$A : \{a_1, \ldots, a_m\}$, $B = \{b_1, \ldots, b_n\}$
$R \subseteq A \times B$;
$R_i = \{b \in B \mid (a_i, b) \in R\}$, $S_i = \{a \in A \mid (a, b_i) \in R\}$
Dann erhalten wir

$$|R| = \sum_{i=1}^{n} |R_i| = \sum_{j=1}^{m} |S_j|.$$

Beweis: Setze

$$x_{ij} = \begin{cases} 1 & \text{falls } (a_i, b_j) \in R \\ 0 & \text{sonst.} \end{cases}$$

Beispiel: $\tau(n) =$ mittlere Anzahl der Teiler von x mit $1 \leq x \leq n$, also $\tau(n) = \frac{1}{n}\sum_{i=1}^{n} d(i)$, wobei $d(i)$ =Anzahl der Teiler von i.

$n = 6$: $A = B = \{1, 2, 3, 4, 5, 6\}$,

$R \subseteq A \times B$, $aRb :\Leftrightarrow a \mid b$.

R	1	2	3	4	5	6
1	1	1	1	1	1	1
2	0	1	0	1	0	1
3	0	0	1	0	0	1
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

p prim: $d(p) = 2$, $d(p^e) = e + 1$,

$$d(p_1^{e_1} \cdots p_k^{e_k}) = (e_1 + 1) \cdots (e_k + 1).$$

Kombinatorische Grundprobleme

$S_i = \{a \in A \mid aRi\}$ ... Teiler von i
$R_j = \{b \in B \mid jRb\}$ ... Vielfache von j.

$$\begin{aligned}
\tau(n) &= \frac{1}{n}\sum_{i=1}^{n} d(i) = \frac{1}{n}\sum_{i=1}^{n} |S_i| = \frac{1}{n}\sum_{j=1}^{n} |R_j| \\
&= \frac{1}{n}\sum_{j=1}^{n} \left\lfloor \frac{n}{j} \right\rfloor = \frac{1}{n}\sum_{j=1}^{n} \frac{n}{j} + \mathcal{O}(1) \\
&\sim \log n
\end{aligned}$$

•) Das Schubfachprinzip

Falls $|A| > |B|$ und $f : A \to B$, dann ist f nicht injektiv.

Allgemeiner:
Seien $A_1, A_2, \ldots, A_k$ endliche, paarweise disjunkte Mengen und $|A_1 \cup A_2 \cup \cdots \cup A_k| > kr$, dann gibt es einen Index i mit $|A_i| > r$.

Beispiel: Es gibt mindestens 2 Menschen in Österreich, die am selben Tag in der gleichen Stunde geboren wurden.

Beispiel:

Für jede ungerade Zahl q gibt es ein ganzzahliges Vielfaches der Form $2^i - 1$.

Beweis: O.B.d.A. gelte $q > 0$. Sei $a_i = 2^i - 1$.

Wir betrachten $a_1, a_2, \ldots, a_q \mod q$.

Falls für $a_i \equiv 0$ für ein i, dann sind wir fertig.

Andernfalls gibt es i, j $(i < j)$, sodass $a_i \equiv a_j \mod q$.

Daraus folgt $a_i - a_j = qa$ mit $a \in \mathbb{Z}$.

Aber $a_i - a_j = 2^i(1 - 2^{j-i})$ und somit ist q ein Teiler von $(2^{j-i} - 1) = a_{j-i}$ ↯

•) Das Inklusions-Exklusions-Prinzip

Sei A eine endliche Menge, und $E_1, \ldots, E_m$ Eigenschaften.
Wir setzen $A_i = \{x \in A \mid x \text{ hat die Eigenschaft } E_i\}$.

Wie viele Elemente der Menge A besitzen keine der Eigenschaften $E_1, \ldots, E_m$?

$$\left| A \setminus \bigcup_{i=1}^{m} A_i \right| = |A| + \sum_{\emptyset \neq \mathcal{I} \subseteq \{1,2,\ldots,m\}} (-1)^{|\mathcal{I}|} \left| \bigcap_{i \in \mathcal{I}} A_i \right|$$

$$\left| \bigcup_{i=1}^{m} A_i \right| = \sum_{\emptyset \neq \mathcal{I} \subseteq \{1,\ldots,m\}} (-1)^{|\mathcal{I}|+1} \left| \bigcap_{i \in \mathcal{I}} A_i \right|$$

Spezialfälle:

$$\begin{aligned}
|A \cup B| &= |A| + |B| - |A \cap B| \\
|A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| \\
&\quad + |A \cap B \cap C|
\end{aligned}$$

4) Divide & Conquer

Matrizenmultiplikation

Seien $A = (a_{i,j})_{i,j=1}^n$, $B = (b_{i,j})_{i,j=1}^n$ quadratische Matrizen und $C = AB = (c_{i,j})_{i,j=1}^n$ mit $c_{i,j} = \sum_{k=1}^n a_{i,k} b_{k,j}$.

Algorithm SQUARE-MATRIX-MULTIPLY(A, B)

```
n := Anzahl der Zeilen von A
Sei C eine neue n × n-Matrix
for i = 1 to n do
    for j = 1 to n do
        c_ij := 0
        for k = 1 to n do
            c_ij := c_ij + a_ik b_kj
        end for
    end for
end for
return C
```

3 Schleifen mit je n Iterationen, daher Laufzeit $T(n) = \Theta(n^3)$

Divide & Conquer

Sei $n = 2^k$. Teile A und B:

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}, \quad B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix}, \quad C = AB = \begin{pmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{pmatrix}.$$

$$C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j} \quad i,j = 1,2.$$

Algorithm SQU-MAT-MUL-REC(A, B)

$n :=$ Anzahl der Zeilen von A
Sei C eine neue $n \times n$-Matrix
if $n = 1$ **then**
    $C_{11} := A_{11}B_{11}$
else
    partitioniere A, B und C wie oben in 4 Teilmatrizen
    C_{11} :=SQU-MAT-MUL-REC($A_{1,1}, B_{1,1}$)+SQU-MAT-MUL-REC($A_{1,2}, B_{2,1}$)
    C_{12} :=SQU-MAT-MUL-REC($A_{1,1}, B_{1,2}$)+SQU-MAT-MUL-REC($A_{1,2}, B_{2,2}$)
    C_{21} :=SQU-MAT-MUL-REC($A_{2,1}, B_{1,1}$)+SQU-MAT-MUL-REC($A_{2,2}, B_{2,1}$)
    C_{22} :=SQU-MAT-MUL-REC($A_{2,1}, B_{1,2}$)+SQU-MAT-MUL-REC($A_{2,2}, B_{2,2}$)
end if
return C

Algorithm SQU-MAT-MUL-REC(A, B)

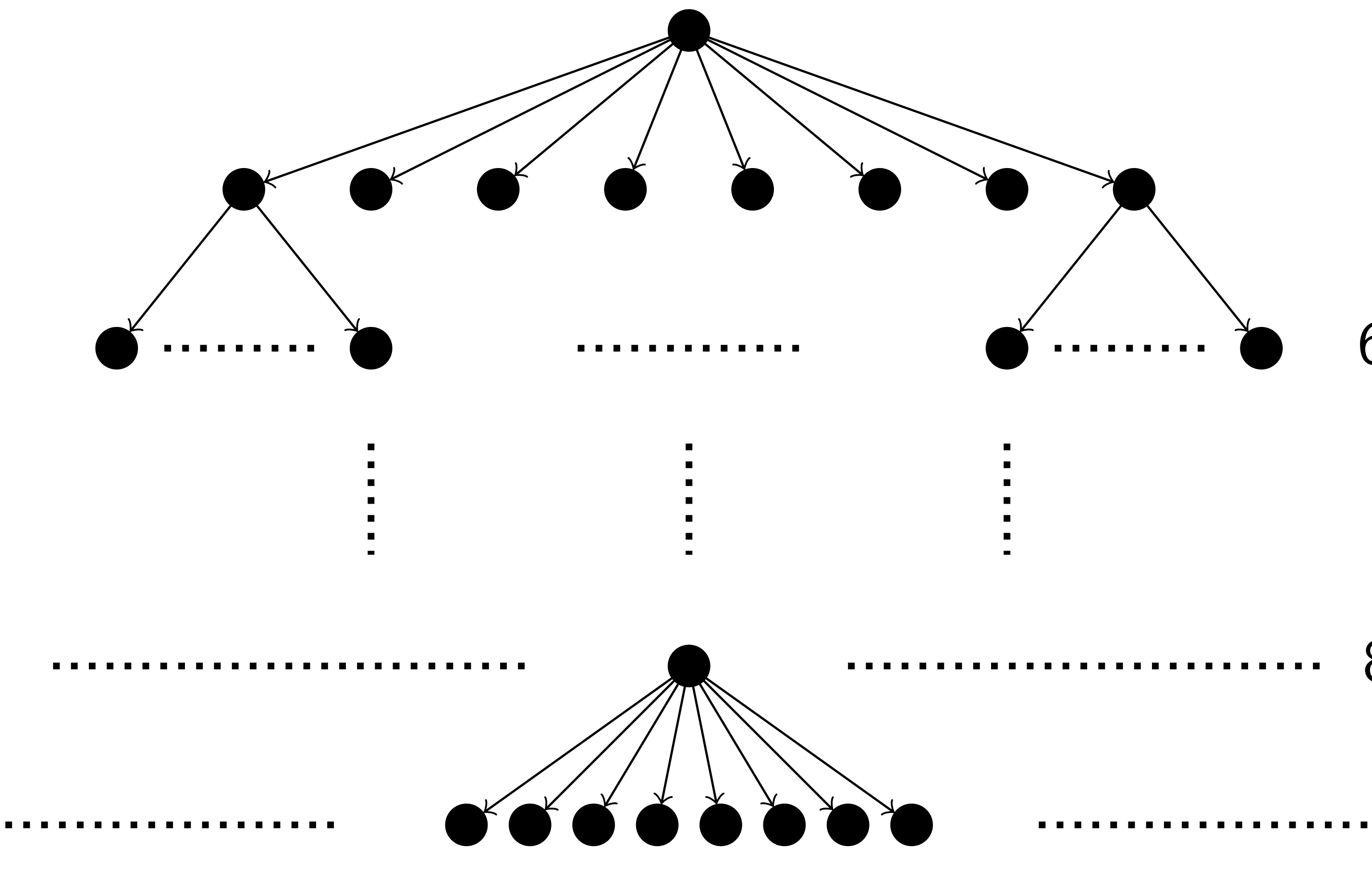
```
n = # Zeilen von A
Sei C eine neue n × n-Matrix
if n = 1 then
    C11 = A11 B11
else
    partitioniere A, B und C wie oben in 4 Teilmatrizen
    C11 =SQU-MAT-MUL-REC(A1,1, B1,1)+SQU-MAT-MUL-REC(A1,2, B2,1)
    C12 =SQU-MAT-MUL-REC(A1,1, B1,2)+SQU-MAT-MUL-REC(A1,2, B2,2)
    C21 =SQU-MAT-MUL-REC(A2,1, B1,1)+SQU-MAT-MUL-REC(A2,2, B2,1)
    C22 =SQU-MAT-MUL-REC(A2,1, B1,2)+SQU-MAT-MUL-REC(A2,2, B2,2)
end if
return C
```

$$T(n) = \begin{cases} \Theta(1) & \text{für } n = 1, \\ 8T\left(\frac{n}{2}\right) + \Theta(1) + \Theta(n^2) & \text{sonst.} \end{cases}$$

Interpretation als Rekursionsbaum

Sei $n = 2^k$ und

$$T(n) = 8\,T\left(\frac{n}{2}\right) + \Theta(n^2),\ n \geq 2, \quad T(1) = \Theta(1).$$



$$\Theta\left(n^2\right)$$

$$8\Theta\left(\frac{n^2}{4}\right)$$

$$64\Theta\left(\frac{n^2}{16}\right)$$

$$8^\ell\Theta\left(\frac{n^2}{4^\ell}\right)$$

$$8^k\Theta(1)$$

Divide & Conquer

$$T(n) = \begin{cases} \Theta(1) & \text{für } n = 1, \\ 8\,T\left(\frac{n}{2}\right) + \Theta(1) + \Theta(n^2) & \text{sonst.} \end{cases}$$

Lösen für $n = 2^k$:

$$T(2^k) = \Theta(2^{2k}) + 8\,T(2^{k-1}) = \Theta(2^{2k}) + 8\Theta(2^{2k-2}) + 64\,T(2^{k-2})$$

$$= \sum_{\ell=0}^{k} 8^\ell \Theta(2^{2(k-\ell)}) = \sum_{\ell=0}^{k} \Theta(2^{2k+\ell})$$

$$= \Theta\left(2^{2k}\frac{2^{k+1}-1}{2-1}\right) = \Theta(2^{3k}) = \Theta(n^3)$$

Der Strassen-Algorithmus für die Matrizenmultiplikation

Sei $n = 2^k$. Idee:

(1) Partitioniere A, B, C in je 4 Teilmatrizen

(2) Konstruiere 10 $\frac{n}{2} \times \frac{n}{2}$-Matrizen $S_1, \ldots, S_{10}$, die Summen bzw. Differenzen von Matrizen aus dem ersten Schritt sind.

(3) Berechne rekursiv 7 Matrizenprodukte $P_1, \ldots, P_7$ mittels der in (1) und (2) erzeugten Matrizen.

(4) $C_{i,j}$ entsteht durch geeignete Linearkombinationen der P_i.

Aufwand: (1): $\Theta(1)$, (2): $\Theta(n^2)$, (3): $7T\left(\frac{n}{2}\right)$, (4): $\Theta(n^2)$.

Also

$$T(n) = \begin{cases} \Theta(1) & \text{für } n = 1, \\ 7T\left(\frac{n}{2}\right) + \Theta(n^2) & \text{sonst.} \end{cases}$$

Divide & Conquer

Algorithm STRASSEN(A, B)

$n :=$ Anzahl der Zeilen von A
Sei C eine neue $n \times n$-Matrix
if $n = 1$ **then**
  $C_{11} := A_{11} B_{11}$
else partitioniere A, B und C in 4 Teilmatrizen
  $S_1 := B_{1,2} - B_{2,2}$; $S_2 := A_{1,1} + A_{1,2}$
  $S_3 := A_{2,1} + A_{2,2}$; $S_4 := B_{2,1} - B_{1,1}$
  $S_5 := A_{1,1} + B_{2,2}$; $S_6 := B_{1,1} + B_{2,2}$
  $S_7 := A_{1,1} - A_{2,2}$; $S_8 := B_{2,1} + B_{2,2}$
  $S_9 := A_{1,1} - A_{2,1}$; $S_{10} := B_{1,1} + B_{2,2}$
  $P_1 :=$STRASSEN$(A_{1,1}, S_1)$
  $P_2 :=$STRASSEN$(S_2, B_{2,2})$
  $P_3 :=$STRASSEN$(S_3, B_{1,1})$
  $P_4 :=$STRASSEN$(A_{2,2}, S_4)$
  $P_5 :=$STRASSEN(S_5, S_6)
  $P_6 :=$STRASSEN(S_7, S_8)
  $P_7 :=$STRASSEN(S_9, S_{10})
  $C_{11} := P_4 + P_5 + P_6 - P_2$; $C_{12} := P_1 + P_2$
  $C_{21} := P_3 + P_4$; $C_{22} := P_1 + P_5 - P_3 - P_7$
end if
return C

Divide & Conquer

Korrektheit: Nachrechnen
Laufzeit: Sei $n = 2^k$.

$$T(n) = \begin{cases} \Theta(1) & \text{für } n = 1, \\ 7T\left(\frac{n}{2}\right) + \Theta(n^2) & \text{sonst.} \end{cases}$$

Daraus folgt:

$$\begin{aligned} T\left(2^k\right) &= \sum_{\ell=0}^{k} 7^\ell \Theta\left(\frac{2^k}{2^\ell}\right)^2 = \sum_{\ell=0}^{k} \Theta\left(2^{2k+\ell(\log_2 7-2)}\right) \\ &= \Theta\left(2^{2k} \sum_{\ell=0}^{k} 2^{(\log_2 7-2)\ell}\right) = \Theta\left(2^{2k+(\log_2 7-2)k}\right) \\ &= \Theta\left(n^{\log_2 7}\right) \end{aligned}$$

Anmerkung: $\log_2 7 \approx 2.807$

Frage: Matrixmultiplikation in $\mathcal{O}\left(n^{\omega+\varepsilon}\right)$ möglich; $\omega_{\min} = ?$
Vermutung: $\omega = 2$

- Coppersmith & Winograd 1987: Modifikation v. Strassens Algorithmus für Trilinearformen, basierend auf Berechnungen der Bewertung der m-ten Tensorpotenz einer speziellen Trilinearform $\leadsto \omega < 2,38719$.
- Coppersmith & Winograd 1990: Feinere Analyse des Tensorquadrates der Trilinearform $\leadsto \omega < 2,375477$.
- Slothers 2010, Davie & Slothers 2013: Erweiterung des Ansatz von Coppersmith & Winograd auf die dritte Tensorpotenz $\leadsto$ keine Verbesserung; vierte Tensorpotenz $\leadsto \omega < 2,3736898$ durch numerische Lösung des zugrunde liegenden Optimierungsproblems.
- Vassilevska-Williams 2012: Numerische Lösung nicht optimal, bessere Lösung $\leadsto \omega < 2,3729269$. Rekursive Analyse der Tensorpotenzen (bis zur 8. Potenz): $\leadsto \omega < 2,3729$.
- LeGall 2014: Modifikation des Ansatzes $\leadsto$ Analyse von Tensorpotenzen bis zu polynomieller Ordnung $\leadsto \omega < 2,3728639$.
- Alman und Vassilevska-Williams 2021: weitere Modifikationen $\leadsto \omega < 2,37286$.

Die Substitutionsmethode

Guess & Prove – Zugang

Beispiel: $T(n) = T(n-1) + n, \quad n \geq 1, \quad T(0) = 0, T(1) = 1$

Vermutung: $T(n) = \Theta(n^2)$, d.h. zu zeigen ist $T(n) \leq Cn^2$ und $T(n) \geq cn^2$

$$T(n) \leq C(n-1)^2 + n = Cn^2 - (2C-1)n + C$$

$$\underset{C=1}{=} n^2 - n + 1 \leq n^2 = Cn^2$$

$$T(n) \geq c(n-1)^2 + n = cn^2 - (2c-1)n + c$$

$$\underset{c=1/4}{=} \frac{n^2}{4} + \frac{n}{2} + \frac{1}{4} \geq \frac{n^2}{4} = cn^2$$

Analog: Erraten der exakten Form $T(n) = an^2 + bn + c$.

Rechnen ergibt $T(n) = \frac{n^2}{2} + \frac{n}{2}$.

Divide & Conquer

Beispiel: $T(n) = T\left(\lfloor \frac{n}{2} \rfloor\right) + T(\lceil \frac{n}{2} \rceil) + 1$

Vermutung: $T(n) = \mathcal{O}(n)$, d.h. $T(n) \leq cn$

$$T(n) \leq c\left\lfloor \frac{n}{2} \right\rfloor + c\left\lceil \frac{n}{2} \right\rceil + 1 = cn + 1 > cn \quad \lightning$$

$T(n) = \mathcal{O}\left(n^2\right)$ funktioniert, aber $\mathcal{O}\left(n\right)$ stimmt.
Versuche Ansatz $T(n) \leq cn - d$

$$T(n) \leq c\left\lfloor \frac{n}{2} \right\rfloor - d + c\left\lceil \frac{n}{2} \right\rceil - d + 1 = cn - 2d + 1 \leq cn - d \text{ falls } d \geq 1.$$

Divide & Conquer

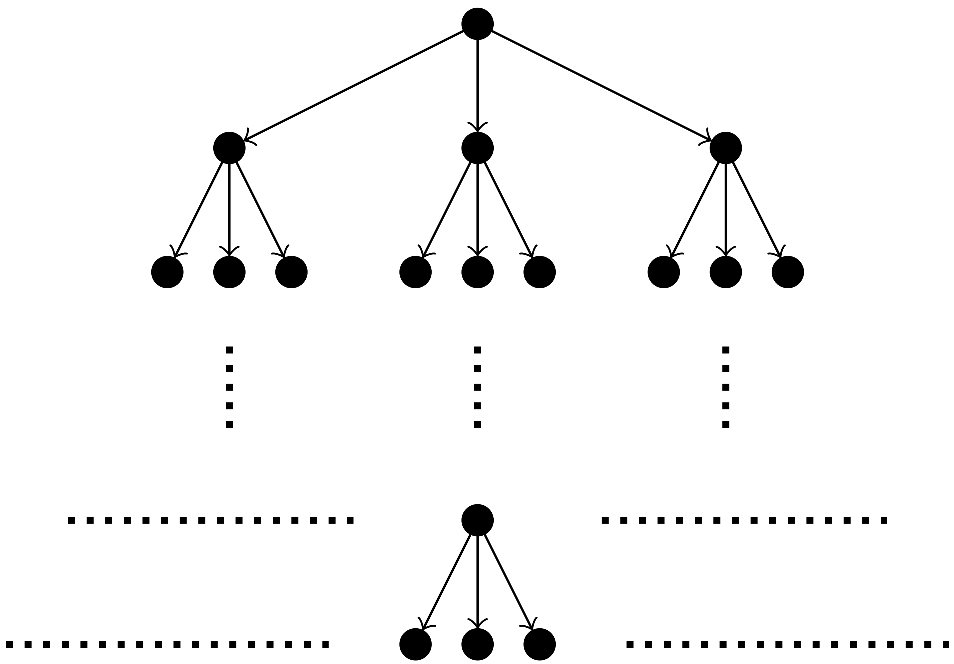
Die Rekursionsbaummethode

Beispiel: Sei $n = 4^k$ und

$$T(n) = 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + \Theta(n^3), \quad T(1) = \Theta(1).$$

Vereinfachen zu

$$T(n) = 3T\left(\frac{n}{4}\right) + \tilde{c}n^3, \quad T(1) = c.$$



$$cn^3$$

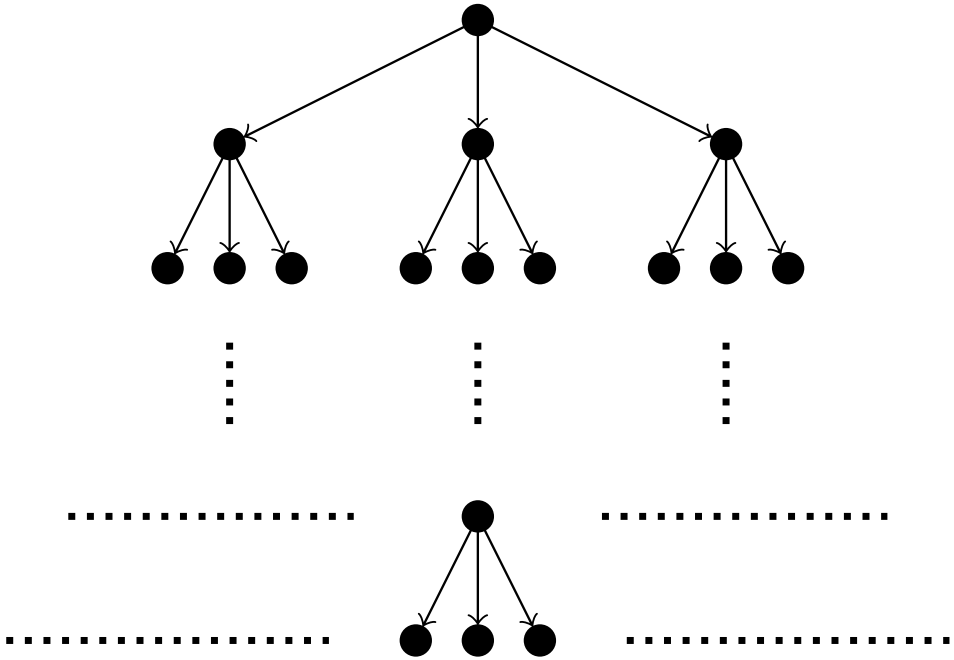
$$\frac{3}{64}cn^3$$

$$\frac{9}{64^2}cn^3$$

$$\frac{3^\ell}{64^\ell}cn^3$$

$$3^{\log_4 n}\tilde{c} = \tilde{c}n^{\log_4 3}$$

Divide & Conquer



$$cn^3$$

$$\frac{3}{64}cn^3$$

$$\frac{9}{64^2}cn^3$$

$$\frac{3^\ell}{64^\ell}cn^3$$

$$3^{\log_4 n}\tilde{c} = \tilde{c}n^{\log_4 3}$$

$$T(n) = cn^3 \sum_{\ell=0}^{\log_4 n - 1} \left(\frac{3}{64}\right)^\ell + \Theta\left(n^{\log_4 3}\right)$$

$$\leq cn^3 \sum_{\ell \geq 0} \left(\frac{3}{64}\right)^\ell + \Theta\left(n^{\log_4 3}\right) = \mathcal{O}\left(n^3\right)$$

Zurück zu

$$T(n) = 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + \Theta(n^3), \quad T(1) = \Theta(1).$$

Vermutung: $T(n) = \Theta(n^3)$

Beweis mit Substitutionsmethode:

$$\begin{aligned} T(n) &\leq 3c\left\lfloor \frac{n}{4} \right\rfloor^3 + dn^3 \leq 3c\frac{n^3}{64} + dn^3 \\ &= \frac{3}{64}cn^3 + dn^3 = \left(\frac{3}{64}c + d\right) n^3 \end{aligned}$$

beschränkt durch cn^3, falls $c \geq \frac{64}{61}d$.

$T(n) = \Omega(n^3)$ klar. $\square$

Das Mastertheorem

Satz (Mastertheorem)

Gegeben seien $a \geq 1, b > 1$, *Funktionen* $f : \mathbb{R}_0^+ \to \mathbb{R}_0^+, T : \mathbb{N} \to \mathbb{R}$ *und* $\varepsilon > 0$.
$T(n)$ *erfülle die Rekursion* $T(n) = aT\left(\frac{n}{b}\right) + f(n)$
($\frac{n}{b}$ *kann auch für* $\left\lfloor \frac{n}{b} \right\rfloor$ *oder* $\left\lceil \frac{n}{b} \right\rceil$ *stehen).*
Dann gilt:

1. *Falls* $f(n) = \mathcal{O}\left(n^{\log_b a - \epsilon}\right)$, *dann gilt* $T(n) = \Theta\left(n^{\log_b a}\right)$;
2. *falls* $f(n) = \Theta\left(n^{\log_b a}\right)$, *dann gilt* $T(n) = \Theta\left(n^{\log_b a} \log n\right)$;
3. *falls* $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ *und es ein* $c < 1$ *gibt, sodass für hinreichend große* n *die Ungleichung* $af\left(\frac{n}{b}\right) \leq cf(n)$ *erfüllt ist, dann gilt* $T(n) = \Theta(f(n))$.

Divide & Conquer

1. Falls $f(n) = \mathcal{O}\left(n^{\log_b a - \epsilon}\right)$, dann gilt $T(n) = \Theta\left(n^{\log_b a}\right)$;
2. falls $f(n) = \Theta\left(n^{\log_b a}\right)$, dann gilt $T(n) = \Theta\left(n^{\log_b a} \log n\right)$;
3. falls $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ und es ein $c < 1$ gibt, sodass für hinreichend große n die Ungleichung $af(\frac{n}{b}) \leq cf(n)$ erfüllt ist, dann gilt $T(n) = \Theta(f(n))$.

Beispiele:

- $T(n) = 9T\left(\frac{n}{3}\right) + n$, $\log_3 9 = 2$, $f(n) = n = \mathcal{O}\left(n^{2-\varepsilon}\right)$
  Daher gilt nach Fall 1: $T(n) = \Theta(n^2)$.
- $T(n) = T\left(\frac{n}{4}\right) + 1$, $\log_4 1 = 0$, $f(n) = \Theta(1)$.
  Daher gilt nach Fall 2: $T(n) = \Theta(\log n)$.
- $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$,
  $\log_2 2 = 1$, $f(n) = n \log n$, aber $n^{\log_b a} = n$
  Mastertheorem nicht anwendbar!

Beweis des Mastertheorems

Sei zunächst $n = b^k$ $(k \in \mathbb{N})$

Lemma 1

Es gelte

$$T(n) = \begin{cases} aT\left(\frac{n}{b}\right) + f(n) & \text{für } n = b^k \text{ mit } k > 0, \\ \Theta(1) & \text{für } n = 1. \end{cases}$$

Dann gilt

$$T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{\ell=0}^{\log_b n - 1} a^\ell f\left(\frac{n}{b^\ell}\right).$$

Beweis: Sukzessives Einsetzen bzw. Rekursionsbaummethode. $\square$

Lemma 2

$a \geq 1$, $b > 1$, $n = b^k$, $f \geq 0$, $g(n) = \sum_{\ell=0}^{k-1} a^\ell f\left(\frac{n}{b^\ell}\right)$. Dann gilt:

1. *Falls $f(n) = \mathcal{O}\left(n^{\log_b a - \epsilon}\right)$, dann gilt $g(n) = \mathcal{O}\left(n^{\log_b a}\right)$;*
2. *falls $f(n) = \Theta\left(n^{\log_b a}\right)$, dann gilt $g(n) = \Theta\left(n^{\log_b a} \log n\right)$;*
3. *falls $f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$ und es ein $c < 1$ gibt, sodass $af(\frac{n}{b}) \leq cf(n)$, dann gilt $g(n) = \Theta(f(n))$.*

Beweis: 1.Fall: $g(n) = \mathcal{O}(h(n))$ mit

$$h(n) = \sum_{\ell=0}^{k-1} a^\ell \left(\frac{n}{b^\ell}\right)^{\log_b a - \varepsilon} = n^{\log_b a - \varepsilon} \sum_{\ell=0}^{k-1} \left(\frac{a}{b^{\log_b a - \varepsilon}}\right)^\ell$$

$$= n^{\log_b a - \varepsilon} \sum_{\ell=0}^{k-1} b^{\varepsilon \ell} = n^{\log_b a - \varepsilon} \frac{b^{\varepsilon \log_b n} - 1}{b - 1} = \mathcal{O}\left(n^{\log_b a}\right)$$