Datenstrukturen, Algorithmen und Programmierung 2

Amin Coja-Oghlan

May 9, 2023

Lehrstuhl Informatik 2
Fakultät für Informatik

Rekurrenzen

Motivation

- bei der Analyse von Algorithmen begegnen häufig Rekurrenzen
- beispielsweise erfüllt die Laufzeit von `Quicksort` mit dem `Select`-Algorithmus die Rekurrenz

$$\mathcal{T}_n = O(n) + 2\mathcal{T}_{\lfloor n/2 \rfloor}$$

- das *Master-Theorem* erlaubt es, Rekurrenzen bequem abzuschätzen

Rekurrenzen

Master-Theorem

Seien $a \geq 1$, $b > 1$ reelle Zahlen und seien $T : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ und $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ Funktionen, so daß

$$T(x) = aT(x/b) + f(x) \qquad \text{für alle } x \in \mathbb{R}_{\geq 0}.$$

(i) falls $f(x) = O(x^{\log_b(a)-\varepsilon})$ für ein $\varepsilon > 0$, dann gilt $T(x) = \Theta(x^{\log_b a})$.

(ii) falls $f(x) = \Theta(x^{\log_b a} \log^k x)$ für ein $k \geq 0$, dann gilt $T(x) = \Theta(x^{\log_b a} \log^{k+1} x)$.

(iii) falls $f(x) = \Omega(x^{\log_b a+\varepsilon})$ für ein $\varepsilon > 0$ und falls außerdem Zahlen $c < 1$, $x_0 > 0$ existieren, so daß $af(x/b) \leq cf(x)$ für alle $x > x_0$, dann gilt $T(x) = \Theta(f(x))$.

Rekurrenzen

Lemma

Seien $a \geq 1$ und $b > 1$ reelle Zahlen, sei $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ eine Funktion und sei $T : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ eine Funktion, so daß

$$T(x) = \begin{cases} \Theta(1) & \text{falls } 0 \leq x \leq 1, \\ aT(x/b) + f(x) & \text{sonst.} \end{cases} \tag{1}$$

Dann gilt

$$T(x) = \Theta(x^{\log_b a}) + \sum_{0 \leq j \leq \log_b x} a^j f(x/b^j). \tag{2}$$

Rekurrenzen

Lemma

Seien $a \geq 1$ und $b > 1$ reelle Zahlen, sei $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ eine und sei

$$g(x) = \sum_{0 \leq j \leq \log_b x} a^j f(x/b^j) \qquad (x \geq 1). \tag{3}$$

(i) falls $f(x) = O(x^{\log_b(a)-\varepsilon})$ für ein $\varepsilon > 0$, dann gilt $g(x) = \Theta(x^{\log_b a})$.

(ii) falls $f(x) = \Theta(x^{\log_b a} \log^k x)$ für ein $k \geq 0$, dann gilt $g(x) = \Theta(x^{\log_b a} \log^{k+1} x)$.

(iii) falls $f(x) = \Omega(x^{\log_b a+\varepsilon})$ für ein $\varepsilon > 0$ und falls außerdem Zahlen $c < 1$, $x_0 > 0$ existieren, so daß $af(x/b) \leq cf(x)$ für alle $x > x_0$, dann gilt $g(x) = \Theta(f(x))$.

Rekurrenzen

Matrixmultiplikation

- für zwei $n \times n$-Matrizen $A = (a_{ij})_{i,j=1,\ldots,n}$ and $B = (b_{ij})_{i,j=1,\ldots,n}$ möchten wir das Produkt $C = A \cdot B$ mit Einträgen

$$c_{ij} = \sum_{\ell=1}^{n} a_{i\ell} b_{\ell j} \qquad (i, j = 1, \ldots, n)$$

  berechnen.
- der naive Algorithmus benötigt $O(n^3)$ Multiplikationen

Rekurrenzen

Strassen-Algorithmus

- wir nehmen an, daß $n = 2^t$ für $t \in \mathbb{N}$
- *divide & conquer*: zerlege A, B, C in Untermatrizen der Größe $n/2 \times n/2$:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \qquad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Rekurrenzen

Strassen-Algorithmus

- nun berechne rekursiv die sieben Produkte

$$
\begin{aligned}
P_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) \\
P_2 &= (A_{21} + A_{22})B_{11} \\
P_3 &= A_{11}(B_{12} - B_{22}) \\
P_4 &= A_{22}(B_{21} - B_{11}) \\
P_5 &= (A_{11} + A_{12})B_{22} \\
P_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\
P_7 &= (A_{12} - A_{22})(B_{21} + B_{22})
\end{aligned}
$$

Rekurrenzen

Strassen-Algorithmus

- dann erhalten wir

$$
\begin{aligned}
C_{11} &= P_1 + P_4 - P_5 + P_7 \\
C_{12} &= P_3 + P_5 \\
C_{21} &= P_2 + P_4 \\
C_{22} &= P_1 + P_3 - P_2 + P_6
\end{aligned}
$$

Rekurrenzen

Satz

Der Strassen-Algorithmus benötigt $T(n) = \Theta(n^{\log_2 7})$ Rechenoperationen.

Beweis

- die Laufzeit erfüllt die Rekurrenz

$$T(n) = 7T(n/2) + O(n^2).$$

- daher folgt die Behauptung aus dem Master-Theorem.

Rekurrenzen

Zusammenfassung

- das Master-Theorem ermöglicht die Abschätzung von Funktionen, die durch Rekurrenzen definiert sind
- das Theorem gilt entsprechend für Funktionen $T : \mathbb{N} \to \mathbb{R}_{\geq 0}$, wenn in der Rekurrenz $\lfloor \cdot \rfloor$ oder $\lceil \cdot \rceil$-Symbole auftreten
- *nicht alle Rekurrenzen lassen sich mit dem Master-Theorem lösen*
- *Anwendung:* Strassen-Algorithmus