Effiziente Algorithmen SS 19

Dr. Gengler

Aufgabenblatt 2

(Besprechung am 24.04.2019)

Master-Theorem

Seien $a \geq 1$, $b > 1$, $f : \mathbb{N} \to \mathbb{N}$:und genüge T der Rekursion $T(n) = a \cdot T(\frac{n}{b}) + f(n)$ für alle n.

1. Falls $f(n) \in \mathcal{O}(n^{\log_b a - \varepsilon})$ für ein $\varepsilon > 0$, dann gilt: $T(n) \in \Theta(n^{\log_b a})$
2. Falls $f(n) = \Theta(n^{\log_b a})$, dann gilt: $T(n) = \Theta(n^{\log_b a} \cdot \log n) = \Theta(f(n) \cdot \log n)$
3. Falls $f(n) = \Omega(n^{\log_b a + \varepsilon})$ für ein $\varepsilon > 0$ und wenn $a \cdot T(\frac{n}{b}) \leq c \cdot f(n)$ für ein $c < 1$ und n groß genug, dann gilt: $T(n) = \Theta(f(n))$

Aufgabe 1
Für welche der folgenden Rekursionsgleichungen ist das Master-Theorem anwendbar? Falls anwendbar, so lösen Sie die Rekursionsgleichung.

1. $T(n) = 2 \cdot T(\frac{n}{2}) + n$ für alle $n > 0$ und $T(0) = 2$
2. $T(n) = 4 \cdot T(\frac{n}{2}) + n$ für alle $n > 0$ und $T(0) = 2$
3. $T(n) = 3 \cdot T(\frac{n}{2}) + 2n$ für alle $n > 0$ und $T(0) = 2$
4. $T(n) = 8 \cdot T(\frac{n}{2}) + n^2$ für alle $n > 0$ und $T(0) = 4$
5. $T(n) = 7 \cdot T(\frac{n}{2}) + n^2$ für alle $n > 0$ und $T(0) = 4$
6. $T(n) = 3 \cdot T(\frac{n}{4}) + cn^2$ für alle $n > 0$ und ein $c > 1$ und $T(0) = 2$
7. $T(n) = T(\frac{2}{3}n) + 1$ für alle $n > 0$ und $T(0) = 2$
8. $T(n) = 2 \cdot T(\frac{n}{2}) + n \log n$ für alle $n > 0$ und $T(0) = 2$

Aufgabe 2
Schlagen Sie die Logarithmenregeln nach.

Aufgabe 3
Wir betrachten die Rekursionsgleichung $T(n) = 2 \cdot T(n/2) + n - 1$ mit den Startwerten $T(1) = 0$ und $T(2) = 1$. Zeigen Sie per Induktion, dass es ein $c \in \mathbb{N}$ gibt mit $T(n) \leq c \cdot n \cdot \log n$.

Aufgabe 4
Wir betrachten die Rekursionsgleichung $T(n) = T(n/3) + T(n/2) + 2n$ mit den Startwerten $T(1) = 0$ und $T(2) = 1$. Zeigen Sie, dass $T(n) \in \mathcal{O}(n \log n)$.

Aufgabe 5
Wir betrachten die Fibonacci-Folge $f_0, f_1, f_2, \ldots$ definiert durch das rekursive Bildungsgesetz $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$ und den Anfangswerten $f_0 = 0$ und $f_1 = 1$. Beweisen Sie per Induktion:

$$f_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right]$$

Aufgabe 6
Erhält man einen besseren Algorithmus, wenn nan Merge-Sort so abändert, dass die zu sortierende Liste statt in zwei gleichgroße Listen in vier gleichgroße Listen unterteilt wird, die dann rekursiv sortiert werden.

Aufgabe 7
Wir betrachten einen Max-Heap $A[1]...A[n]$ mit $n = 2^k - 1$, so dass für $i = 1, \ldots, 2^{k-1} - 1$ in $A[i]$ der Wert $2^k - i$ gespeichert ist.

Wieviele Möglichkeiten gibt es, die Werte $1, \ldots, 2^{k-1}$ in $A[2^k]...A[2^k - 1]$ so zu speichern, das die Max-Heap-Eigernschaft erfüllt bleibt.

Aufgabe 8
Zeigen Sie:

$$\sum_{i=0}^{\infty} \frac{1}{2^i} = 2$$

Hinweis: Zeigen Sie per Induktion, dass $\sum_{i=0}^{m} \frac{1}{2^i} = 2 - \frac{1}{2^m}$ für alle $m \in \mathbb{N}$ gilt.

Aufgabe 9
Seien $f, g : \mathbb{N} \to \mathbb{R}_+$ totale Funktionen. Zeigen Sie die folgenden Aussagen:

1. $f \in \mathcal{O}(g) \iff \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$
2. $f \in o(g) \iff \limsup_{n\to\infty} \frac{f(n)}{g(n)} = 0$