Effiziente Algorithmen SS 17

Prof. Dr. Meer, Dr. Gengler

Aufgabenblatt 2

Aufgabe 0
Bereiten Sie die Vorlesungen gewissenhaft nach, führen Sie dabei ein Zeitprotokoll. Sollten Sie weniger Zeit für die Nachbereitung verbrauchen als die Vorlesung dauerte, so wiederholen Sie diesen Schritt nochmal.

Aufgabe 1
Schlagen Sie die Logarithmengesetze nach, und behalten Sie diese für den weiteren Verlauf des Studiums präsent.

Aufgabe 2
Arbeiten Sie die Materialien zur Wahrscheinlichkeitsrechnung durch und bearbeiten Sie die darin gestellten Aufgaben.

Aufgabe 3
Schlagen Sie die Logarithmengesetze nach, und behalten Sie diese für den weiteeren Verlauf des Studiums präsent.

Aufgabe 4
Sei $B = (V, E)$ ein (gerichteter) binärer Baum mit Wurzel w. Beweisen oder widerlegen Sie die folgenden Aussagen.

1. Die Höhe des Baumes B beträgt mindestens $\log |V|$.
2. Die Höhe des Baumes B beträgt höchstens $\log |V|$.
3. Mindestens $\frac{|V|}{2}$ Knoten des Baumes B sind Blätter.
4. Höchstens $\frac{|V|}{2}+1$ Knoten des Baumes B sind Blätter.

Aufgabe 5
Zu einer (totalen) Funktion $h : \mathbb{R} \to \mathbb{R}$ definieren wir die Funktion $\underline{h} : \mathbb{N} \to \mathbb{N}$ vermöge $\underline{h}(n) = \lfloor h(n) \rfloor$. Seien nun f und g streng monoton wachsende totale Funktionen von $\mathbb{R}$ nach $\mathbb{R}$. Beweisen oder widerlegen Sie die folgenen Aussagen.

1. $f \in \boldsymbol{o}(g) \quad \Longrightarrow \quad \underline{f^{-1}} \in \boldsymbol{o}(\underline{g^{-1}})$
2. $f \in \boldsymbol{o}(g) \quad \Longrightarrow \quad \underline{g^{-1}} \in \boldsymbol{o}(\underline{f^{-1}})$
3. $f \in \boldsymbol{\mathcal{O}}(g) \quad \Longrightarrow \quad \underline{f^{-1}} \in \boldsymbol{\mathcal{O}}(\underline{g^{-1}})$

Hinweis: Denken Sie zum Beispiel an Exponieren und Logarithmieren.

Aufgabe 6
In der Vorlesung (ausgeteiltes Blatt mit Definitionen) wurden zu einer Funktion $f : \mathbb{N} \to \mathbb{R}_+$ jeweils $\Omega(f)$ und $\Omega_\infty(f)$ definiert. ($\mathbb{R}_+ := \{x \in \mathbb{R} \mid x > 0\}$)

Geben Sie ein Entscheidungsproblem sowie ein zugehöriges Verfahren an Verfahren, so dass die Kostenfunktion des Verfahrens in $\Omega_\infty(f)$ aber nicht in $\Omega(f)$ liegt, wobei $g : \mathbb{N} \to \mathbb{R}$ mit $g(n) = n^2$.

Aufgabe 7
Beweisne Sie die folgenden Aussagen:

$$\begin{array}{llcll}
1. & f \in \mathcal{o}g) & \Longleftrightarrow & 0 = & \sup \frac{f}{g} \\
2. & f \in \mathcal{O}(g) & \Longleftrightarrow & & \sup \frac{f}{g} < \infty \\
3. & f \in \Omega(g) & \Longleftrightarrow & 0 < & \sup \frac{f}{g} \\
4. & f \in \Theta(g) & \Longleftrightarrow & 0 < & \sup \frac{f}{g} < \infty
\end{array}$$

Hierbei steht $\sup \frac{f}{g}$ für $\limsup_{n\to\infty} \frac{f(n)}{g(n)} = x$.

Aufgabe 8
Beweisen oder widerlegen Sie die folgenden Aussagen.

1. $f \in \Omega(g) \implies g \in \mathcal{O}(f)$
2. $f \in \Omega_\infty(g) \implies g \in \mathcal{O}(f)$
3. $\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R}_+ \mid \lim_{n\to\infty} \frac{f(n)}{g(n)} \text{ endlich}\}$
4. $\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R}_+ \mid \limsup_{n\to\infty} \frac{f(n)}{g(n)} \text{ endlich}\}$

Aufgabe 9
Sortiert der folgende Algorithmuss korrekt?

```
     SORT(L)
     IF |L|<=1
          THEN RETURN(L)
Opt1cm     ELSE A:=erstes Element vom L
               L1:= alle Elemente von L kleiner gleich A
               L2:= alle Elemente von L echt größer A
          RETURN(SORT(L1)•SORT(L2));
     FI
```

Aufgabe 10
Geben Sie zu jedem $n \in \mathbb{N}$ mit $n > 10$ Listen an so dass der Insertion-Sort-Algorithmus genau $4n$ bzw. $n^{\frac{3}{2}}$ bzw $n \cdot \text{logn}$ Vertauschungen ausführt.

Aufgabe 11
Sortieren Sie die folgende Liste mittels Heap-Sort.

```
('Agneta','Benny','Björn','Frida','Per','Marie','Kenneth','Nina','Magnus','Thomas')
```

Aufgabe 12
Gegeben ein Heap, dargestellt durch ein Array A. Schreiben Sie ein Programm, das testet, ob ein Wert x im dargestellten Heap vorkommt.

Aufgabe 13
Geben Sie für jedes $n \in \mathbb{N}$ Instanzen der Größe n an, so dass Heap-Sort "möglichst viele" Vergleiche macht.

Aufgabe 14
Schreiben Sie ein Programm, das eine 5-elementige Liste mit höchstens 7 Vergleichen sortiert.

Aufgabe 15
Zeigen Sie, dass für endliche Folgen von natürlichen Zahlen (gegeben in Binärdarstellung, ohne führende Nullen, mit niederwertigstem Bit hinten) Merge-Sort auf einer mehrbändrigen Turing-Maschine in Zeitkomplexität $\mathcal{O}(n \cdot \log n)$ implementiert werden kann. Eine Turingmaschine arbeitet in Zeitkomplexität $\mathcal{O}(n \cdot \log n)$, wenn sie bei einer Inputlänge von n höchstens $\mathcal{O}(n \cdot \log n)$ Schritte macht.

Aufgabe 16
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 17
Zeigen Sie:

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

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

Aufgabe 18 (Regel von L'Hospital)
Schlagen Sie die Aussage, den Beweis und die Anwendungsmöglcihkeiten der Regel von L'Hospital nach. Versuchen Sie die Aufgabe 4 von Blatt 1 mit der Regel von L'Hospital zu lösen.

Aufgabe 19
Seien $f, g : \mathbb{N} \to \mathbb{R}_+$ totale Funktionen. Zeigen Sie die folgenden Aussagen:

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

Aufgabe 20
Beweisne Sie die folgenden Aussagen:

$$\begin{array}{llcll}
1. & f \in o g) & \iff & 0 = & \sup \frac{f}{g} \\
2. & f \in \mathcal{O}(g) & \iff & & \sup \frac{f}{g} < \infty \\
3. & f \in \Omega(g) & \iff & 0 < & \sup \frac{f}{g} \\
4. & f \in \Theta(g) & \iff & 0 < & \sup \frac{f}{g} < \infty
\end{array}$$

Hierbei steht $\sup \frac{f}{g}$ für $\limsup_{n\to\infty} \frac{f(n)}{g(n)} = x$.