

Effiziente Algorithmen I SS 15

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

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

Aufgabe 2

In der Vorlesung (ausgeteiltes Blatt mit Definitionen) wurden zu einer Funktion $f : \mathbb{N} \rightarrow \mathbb{R}_+$ 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 $g : \mathbb{N} \rightarrow \mathbb{R}$ mit $g(n) = n^2$.

Aufgabe 3

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. $\mathcal{O}(g) = \{f : \mathbb{N} \rightarrow \mathbb{R}_+ \mid \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \text{ endlich}\}$
4. $\mathcal{O}(g) = \{f : \mathbb{N} \rightarrow \mathbb{R}_+ \mid \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} \text{ endlich}\}$

Aufgabe 4

Sortiert der folgende Algorithmus korrekt?

```

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

Aufgabe 5

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 $n \cdot \log n$ Vertauschungen ausführt.

Aufgabe 6

Sortieren Sie die folgende Liste mittels Heap-Sort.

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

Aufgabe 7

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

Aufgabe 8

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

Aufgabe 9

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

Aufgabe 10

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 11

Erhält man einen besseren Algorithmus, wenn man 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 12

Zeigen Sie:

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

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

Aufgabe 13 (Regel von L’Hospital)

Schlagen Sie die Aussage, den Beweis und die Anwendungsmöglichkeiten der Regel von L’Hospital nach. Versuchen Sie die Aufgabe 6 von Blatt 1 mit der Regel von L’Hospital zu lösen.

Aufgabe 14

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

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