Datenstrukturen (DS)

Sommersemester 2018

Prof Dr. Ulrich Meyer
Manuel Penschuck
Alex Schickedanz

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
Algorithm Engineering

Übung 0

Ausgabe: 10.04.2018
Abgabe: Präsenzaufgaben

Bei diesem Übungsblatt handelt es sich um eine Präsenzübung. Eine Abgabe ist nicht vorgesehen. Die Aufgaben werden im Zeitraum vom 17.04. bis 30.04. besprochen. Wenn Ihnen noch kein Tutorium zugeteilt wurde, gehen Sie zu einem Tutorium Ihrer Wahl. Sie wechseln damit jedoch nicht das Tutorium!

Aufgabe 0.1. *Fehlerhafte Induktion* (-)

Finden Sie die Fehler in den folgenden Beweisen:

a) Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind. Dazu setzen wir $k = \max\{a, b\}$ und führen eine Induktion nach k. Im Basisschritt haben wir $k = 0$ und deshalb ist $a = 0 = b$ und das war zu zeigen. Im Induktionsschritt ist $\max\{a, b\} = k + 1$, und wir können die Induktionsbehauptung auf $a - 1$ und $b - 1$ anwenden, denn $\max\{a - 1, b - 1\} = k$. Also ist $a - 1 = b - 1$ und die Behauptung $a = b$ folgt.

b) Wir zeigen, dass alle Pferde die gleiche Farbe besitzen und führen einen Beweis durch Induktion über die Zahl k aller Pferde. Im Basisschritt ist $k = 1$ und die Behauptung ist offensichtlich richtig. Für den Induktionsschritt nehmen wir an, dass es $k+1$ Pferde $p_1, \ldots, p_{k+1}$ gibt. Dann haben aber nach Induktionsvoraussetzung $p_1, \ldots, p_k$ die Farbe von p_2 und $p_2, \ldots, p_{k+1}$ ebenfalls die Farbe von p_2 und wir haben die Behauptung gezeigt.

Aufgabe 0.2. *Vollständige Induktion* (-)

Beweisen Sie folgende Formeln und Aussagen durch vollständige Induktion für alle $n \in \mathbb{N}$.

a) $\sum_{i=0}^{n} i \cdot 2^i = (n-1) \cdot 2^{n+1} + 2, \quad$ für $n \geq 0$

b) $\frac{n}{2} < 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{2^n-1} < n, \quad$ für $n \geq 2$

c) $n \cdot \sqrt{n} > n + \sqrt{n}$ für $n \geq 4$

Bitte wenden!

Aufgabe 0.3. (-)

Für ein Problem wurden 5 Algorithmen $A1, \ldots, A5$ mit folgendem Berechnungsaufwand $b_i(n)$ ermittelt, angegeben ist jeweils die Anzahl der Operationen für Eingabelänge n:

$$b_1(n) = n, \quad b_2(n) = 5n^7, \quad b_3(n) = 83n^2, \quad b_4(n) = n \log_2 n, \quad b_5(n) = 2^n$$

Wie verändert sich der Berechnungsaufwand, wenn

a) die Eingabelänge verdoppelt wird?

b) die Eingabelänge um 1 erhöht wird?

Aufgabe 0.4. (-)

Gegeben sind zwölf von 1 bis 12 durchnummerierte Kugeln, von denen 11 gleich schwer sind, und eine leichter oder schwerer ist als die anderen. Es ist bekannt, ob die abweichende Kugel leichter oder schwerer ist. Gesucht ist ein Algorithmus, der mit höchstens 3 Wägungen mit einer Balkenwaage die abweichende Kugel ermittelt. In jede Schale passen bis zu zwölf Kugeln.

Aufgabe 0.5. *Pseudocode* (-)

Beschreiben Sie in Pseudocode das Ermitteln der größten Differenz von $n \geq 1$ Zahlen, die in einem Array A vorliegen.
Die Ausgabe des Algorithmus soll also $\max\{A[i] \;:\; 1 \leq i \leq n\} - \min\{A[i] \;:\; 1 \leq i \leq n\}$ sein.