Algorithmen 1 SS 2013 – Tutorium 7

2. Tutorium

Sarah Lutteropp

30. April 2013

Übersicht

Nachtrag Übungsbetrieb

Bonuspunkte für die Klausur

- $\geq$ 25 % der Übungspunkte: 1 Bonuspunkt
- $\geq$ 50 % der Übungspunkte: 2 Bonuspunkte
- $\geq$ 75 % der Übungspunkte: 3 Bonuspunkte

1. Übungsblatt

Allgemeines zur Korrektur

Ich darf leider keine halben Punkte geben. :-(

Aufgabe 1

- Beweisen, dass etwas nicht gilt: Gegenbeispiel reicht
- 1.d) Ihr dürft o.B.d.A. $f(n) \geq g(n)$ annehmen

Aufgabe 2

Beweis, dass der Algorithmus das richtige tut war nicht gefordert.

Allgemein

- Variablen nicht einfach aus dem Nichts hinschreiben, ohne sie vorher zu definieren!
- Äquivalenzpfeile für Umformungen verwenden

1. Übungsblatt

Möchte jemand vorrechnen?

Abbildung: Vorrechnen für Bonuspunkte

Einfaches Mastertheorem

Gegeben:

$$T(n) = \begin{cases} a & \text{falls } n = 1, \\ c \cdot n + d \cdot T\left(\frac{n}{b}\right) & \text{falls } n > 1 \end{cases}$$

für positive Konstanten a, b, c, d und $n = b^k$ für $k \in \mathbb{N}$

Dann gilt:

$$T(n) = \begin{cases} \Theta(n) & \text{falls } d < b, \\ \Theta(n \log n) & \text{falls } d = b, \\ \Theta(n^{\log_b(d)}) & \text{falls } d > b \end{cases}$$

Allgemeines Mastertheorem

Gegeben:

$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$, mit $a \geq 1, b > 1$ const., $f(n)$ Fkt.

Dann gilt:

1. $f(n) \in O(n^{\log_b(a)-\epsilon}) \Rightarrow T(n) \in \Theta(n^{\log_b(a)})$
2. $f(n) \in \Theta(n^{\log_b(a)}) \Rightarrow T(n) \in \Theta(n^{\log_b(a)} \cdot ld(n))$
3. $f(n) \in \Omega(n^{\log_b(a)+\epsilon})$ und $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$ für $c \in (0,1)$
   $\Rightarrow T(n) \in \Theta(f(n))$

Mastertheorem

Aufgabe

$$T(n) = T(\lceil n/4 \rceil) + T\left(\left\lfloor \frac{3}{4} \cdot n \right\rfloor\right) + n,\ T(1) = 1$$

Versuchen Sie durch Ausfalten des Rekursionsbaumes und Aufsummieren auf ein f mit $T(n) = \Theta(f(n))$ zu kommen.

Mastertheorem

Abgeänderte Aufgabe

$$T(n) = \begin{cases} 1 & \text{für } n \leq 32 \\ T(\lceil n/4 \rceil) + T\left(\left\lfloor \frac{3}{4} \cdot n \right\rfloor + 5\right) + n & \text{für } n > 32 \end{cases}$$

Beweisen Sie, dass auch diese Rekursion in $\Theta(n \log n)$ liegt.

Mastertheorem

Noch eine Aufgabe

$$T(n) = 2T(\lfloor\sqrt{n}\rfloor) + \log(n),\ T(1) = 1$$

Raten Sie ein f mit $T(n) \in O(f(n))$ und beweisen Sie das.

Kreativaufgabe

Algorithmenentwurf

Gegeben sei ein Array $A = A[1], \ldots A[n]$ mit n Zahlen in beliebiger Reihenfolge. Für eine gegebene Zahl x soll ein Paar $(A[i], A[j]), 1 \leq j, j \leq n$ gefunden werden, für das gilt:

$$A[i] + A[j] = x$$

Geben Sie eine Lösung für $x = 33$ und $A = (7, 15, 21, 14, 18, 3, 9)$ an.

Kreativaufgabe

Algorithmenentwurf

Gegeben sei ein Array $A = A[1], \dots A[n]$ mit n Zahlen in beliebiger Reihenfolge. Für eine gegebene Zahl x soll ein Paar $(A[i], A[j]), 1 \leq j, j \leq n$ gefunden werden, für das gilt:

$$A[i] + A[j] = x$$

Geben Sie einen effizienten Algorithmus an, der das Problem in Zeit $\mathcal{O}(n \log n)$ löst, und bei Erfolg ein Paar $(A[i], A[j])$ ausgibt, ansonsten NIL.

Bis zum nächsten Mal! ☺