# Datenstrukturen (DS)

Sommersemester 2018

Prof Dr. Ulrich Meyer
Manuel Penschuck
Alex Schickedanz

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
Algorithm Engineering

# Übung 2

Ausgabe: 01.05.2018
Abgabe: 15.05.2018

Ihre Antworten sind stets zu begründen, solange Sie nicht im Aufgabentext davon befreit werden. Nicht vergessen **Name, Matrikelnummer, Gruppennummer und Veranstaltung** auf die Abgabe zu schreiben und mehrseitige Abgaben zu **tackern**.

**Aufgabe 2.1.** *Master-Theorem* ($5 \cdot 4$ Punkte)

Benutzen Sie das Master-Theorem, um die Laufzeitklasse folgender rekursiver Funktionen $T(n)$ zu berechnen. Nehmen Sie jeweils an, dass $T(1)$ konstant und $n$ eine Potenz von $b$ ist.

a) $T(n) = 7 \cdot T(\frac{n}{3}) + n^2$

b) $T(n) = T(\frac{n}{2}) + n$

c) $T(n) = 5 \cdot T(\frac{n}{2}) + n \log n$

d) $T(n) = 2 \cdot T(\frac{n}{3}) + n^7$

e) $T(n) = 4 \cdot T(\frac{n}{2}) + n^2$

**Aufgabe 2.2.** *Rekursion* ($4 \cdot 7$ Punkte)

Geben Sie für folgende rekursive Gleichungen eine *geschlossene Form* an und beweisen Sie deren Korrektheit mittels Induktion.

a) $T(2) = 2, \quad T(n) = T(n-2) + n \qquad$ für $n > 2$

b) $T(1) = 2, \quad T(n) = T(\frac{n}{2}) + \log_2 n \qquad$ für $n > 1$

c) $T(1) = 2, \quad T(n) = \left(T(\frac{n}{2})\right)^2 \qquad$ für $n > 1$

d) $T(1) = 2, \quad T(n) = n \cdot T(\frac{n}{2}) \qquad$ für $n > 1$

*Hinweis: Für a) kann angenommen werden, dass $n$ gerade ist. Für b) bis d) kann angenommen werden, dass $n$ eine Zweierpotenz ist.*

**Bitte wenden!**

## Aufgabe 2.3. (4 · 8 Punkte)

Wir betrachten folgende Algorithmen für eine ganzzahlige Eingabe $n$:

**Algorithmus A**:

var $x = 0$;
for ($i = 1$ ; $i \leq n$ ; $i$++){
    for ($j = 1$ ; $j \leq f(i)$ ; $j$++){
        $x = x + 1$
    }
}

**Algorithmus B**:

var $y = 1$;
for ($i = 1$ ; $i \leq n$ ; $i$++){
    for ($j = 1$ ; $j \leq g(i)$ ; $j$++){
        $y = 2y$
    }
}

Bestimmen Sie für jede der folgenden Funktionen $f$ bzw. $g$ den Wert der Variable $x$ bzw. $y$ am Ende der Berechnung von Algorithmus A bzw. B.

a) $f(i) = 2^i$

b) $f(i) = n/(2^i)$

c) $g(i) = 3n$

d) $g(i) = n - 2^{\log_2(i)}$

*Hinweis*: Für Aufgabenteil b) kann angenommen werden, dass $n$ eine Zweierpotenz ist.

## Aufgabe 2.4. (20 Punkte)

Ein Roboter befindet sich vor einer unendlich langen Wand. In dieser Wand befindet sich an einer unbekannten Stelle eine Tür, die vom Roboter gefunden werden soll. Dieser kann sich nach links oder rechts bewegen und kann feststellen ob er sich in der aktuellen Position vor der Tür befindet oder nicht.

Entwerfen Sie einen Algorithmus zum Auffinden der Tür, so dass der Roboter eine Strecke der Länge $\mathcal{O}(n)$ zurücklegen muss, wenn $n$ der Abstand vom Startpunkt zur Tür ist. Zeigen Sie sowohl die Korrektheit Ihres Algorithmus als auch die obere Schranke für die zurückgelegte Entfernung.