

# Algorithmen und Datenstrukturen

## Übungsgruppe 14

Utz Pöhlmann

4poehlma@informatik.uni-hamburg.de  
6663579

Louis Kobras

4kobras@informatik.uni-hamburg.de  
6658699

Paul Testa

paul.testa@gmx.de  
6251548

6. November 2015

**Punkte für den Hausaufgabenteil:**

2.1	2.2	2.3	2.4	$\Sigma$

# 1 Zettel vom 14.-16. Oktober – Abgabe: N/A

## 1.1 Präsenzaufgabe 1.1

Wiederholen Sie die  $O$ -Notation und die verwandten Notationen. Wie sind die einzelnen Mengen definiert? Was bedeutet es, wenn  $f \in O(g)$  gilt, was wenn  $f \in \Theta(g)$  gilt und so weiter?

$$\begin{aligned} O(g(n)) : f(n) \in O(g(n)) &\Leftrightarrow \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \|f(n)\| \leq c \cdot \|g(n)\| \\ o(g(n)) : f(n) \in o(g(n)) &\Leftrightarrow \forall c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \|f(n)\| < c \cdot \|g(n)\| \\ \Omega(g(n)) : f(n) \in \Omega(g(n)) &\Leftrightarrow \exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \|f(n)\| \geq c \cdot \|g(n)\| \\ \omega(g(n)) : f(n) \in \omega(g(n)) &\Leftrightarrow \forall c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \|f(n)\| > c \cdot \|g(n)\| \\ \Theta(g(n)) : f(n) \in \Theta(g(n)) &\Leftrightarrow \exists c_1, c_2 \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \forall n \geq n_0 : c_1 \cdot \|g(n)\| \leq \|f(n)\| \leq c_2 \cdot \|g(n)\| \end{aligned}$$

## 1.2 Präsenzaufgabe 1.2

Beweisen Sie:

- $n^2 + 3n - 5 \in O(n^2)$
- $n^2 - 2n \in \Theta(n^2)$
- $n! \in O((n+1)!)$

Gilt im letzten Fall auch  $n! \in o((n+1)!)$ ?

$$\begin{aligned} f(n) \in O(g(n)) &\Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty \\ f(n) &= n^2 + 3n - 5 \\ g(n) &= n^2 \\ \frac{f(n)}{g(n)} &= \frac{n^2 + 3n - 5}{n^2} \end{aligned}$$

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{n^2 + 3n - 5}{n^2} &= \lim_{n \rightarrow \infty} 1 + \frac{3}{n} - \frac{5}{n^2} \\ &= 1 + \frac{3}{\infty} - \frac{5}{\infty^2} \\ &= 1 + 0 + 0 \\ &= 1 < \infty \Rightarrow f(n) \in O(g(n)) \end{aligned}$$

□

$$\begin{aligned} c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N} \forall n \geq n_0 : c_1 \cdot n^2 \leq n^2 - 2n \leq c_2 \cdot n^2 \\ \Leftrightarrow c_1 \leq 1 - \frac{2}{n} \leq c_2 \end{aligned}$$

Dies ist erfüllbar ab  $n_0 \geq 2$ , da für  $n = 1$  im mittleren Ausdruck 0 herauskommt und  $c_1$  größer als 0, aber kleiner als der mittlere Ausdruck sein muss. Ist  $n \geq 2$ , so kommt im mittleren Ausdruck 0,5 heraus, für  $c_1$  lässt sich ein beliebiger Wert aus  $]0; 0.5[$  wählen, sei es an dieser Stelle  $\frac{1}{4}$ . Als Obergrenze für  $c_2$  lässt sich jeder Wert größer oder gleich 1 wählen, da der mittlere Ausdruck nicht größer als 1 werden kann und somit die Bedingung des "kleiner gleichsofort" erfüllt ist.

Somit wird als Ergebnis für die Belegung gewählt:  $c_1 = \frac{1}{4}; c_2 = 1; n_0 = 2$ . Mit dieser Belegung gilt  $n^2 - 2n \in \Theta(n^2)$

□

$$\begin{aligned} f(n) \in O(g(n)) &\Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty \\ f(n) &= n! \\ g(n) &= (n+1)! = (n+1) \cdot n! \end{aligned}$$

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{n!}{(n+1) \cdot n!} &= \lim_{n \rightarrow \infty} \frac{1}{n+1} \\ &= \frac{1}{\infty} \\ &= 0 < \infty \Rightarrow f(n) \in O(g(n)) \end{aligned}$$

Da die Bedingung für  $o(g(n))$  ist, dass der Quotient nicht nur kleiner unendlich, sondern gleich null ist, was hier wie oben gezeigt gegeben ist, gilt auch  $n! \in o((n+1)!)$ .

□

### 1.3 Präsenzaufgabe 1.3

Beweisen oder widerlegen Sie:

1.  $f(n), g(n) \in O(h(n)) \Rightarrow f(n) + g(n) \in O(h(n))$
2.  $f(n), g(n) \in O(h(n)) \Rightarrow f(n) \cdot g(n) \in O(h(n))$

$$\exists c_1 \in \mathbb{R}^+ \exists n_{0_1} \in \mathbb{N} \forall n \geq n_{0_1} : \|f(n)\| \leq c_1 \cdot \|h(n)\|$$

$$\exists c_2 \in \mathbb{R}^+ \exists n_{0_2} \in \mathbb{N} \forall n \geq n_{0_2} : \|g(n)\| \leq c_2 \cdot \|h(n)\|$$

$$n_0 = \max(n_{0_1}, n_{0_2})$$

$$\|f(n) + g(n)\| \leq c_1 \cdot \|h(n)\| + c_2 \cdot \|h(n)\| \leq (c_1 + c_2) \cdot \|h(n)\|$$

Seien  $f(n)$  und  $g(n)$  Polynome zweiten Grades sowie  $h(n)$  ein Polynom dritten Grades. Dann sind sowohl  $f(n)$  als auch  $g(n)$  durch die *limes*-Bedingung in  $O(h(n))$ . Das Produkt zweier Polynome zweiten Grades ist allerdings ein Polynom vierten Grades, sodass gilt:

$$\lim_{n \rightarrow \infty} \frac{n^2 \cdot n^2}{n^3} = \lim_{n \rightarrow \infty} \frac{n^4}{n^3} = \lim_{n \rightarrow \infty} n = \infty$$

Damit ist das Produkt der Polynome nicht mehr in  $O(h(n))$ , da die *limes*-Bedingung, nach der der Quotient der Polynome für  $n$  gegen Unendlich kleiner als Unendlich sein zu hat, nicht erfüllt ist. Damit ist (2) widerlegt.

□

## 2 Zettel vom 15.10. – Abgabe: 26.10.

### 2.1 Übungsaufgabe 2.1

[ | 2 ]

Begründen Sie formal, warum folgende Größenabschätzungen gelten bzw. nicht gelten:

1.  $3n^3 - 6n + 20 \in O(n^3)$
2.  $n^2 \cdot \log n \in O(n^3) \cap \Omega(n^2)$

#### 2.1.1

$$\begin{aligned} 3n^3 - 6n + 20 \in O(n^3) &\Leftrightarrow \lim_{n \rightarrow \infty} \frac{3n^3 - 6n + 20}{n^3} < \infty \\ \lim_{n \rightarrow \infty} \frac{3n^3 - 6n + 20}{n^3} &= \lim_{n \rightarrow \infty} \frac{3n^3}{n^3} - \frac{6n}{n^3} + \frac{20}{n^3} = \lim_{n \rightarrow \infty} 3 - \frac{6}{n^2} + \frac{20}{n^3} = 3 - 0 + 0 < \infty \\ &\Rightarrow 3n^3 - 6n + 20 \in O(n^3) \quad \square \end{aligned}$$

#### 2.1.2

$$\begin{aligned} n^2 \cdot \log n \in O(n^3) \cap \Omega(n^2) &\Leftrightarrow \lim_{n \rightarrow \infty} \frac{n^2 \cdot \log n}{n^3} < \infty \wedge \lim_{n \rightarrow \infty} \frac{n^2 \cdot \log n}{n^2} > 0 \\ \frac{n^2 \cdot \log n}{n^2} &= \frac{1 \cdot \log n}{1} = \log n > 0 \quad \forall n > 1 \Rightarrow n^2 \cdot \log n \in \Omega(n^2) \\ \lim_{n \rightarrow \infty} \frac{n^2 \cdot \log n}{n^3} &= \lim_{n \rightarrow \infty} \frac{\log n}{n} \stackrel{\text{rH}}{=} \lim_{n \rightarrow \infty} \frac{1}{n} \cdot \frac{1}{1} = \lim_{n \rightarrow \infty} \frac{1}{n} = \frac{1}{\infty} = 0 \Rightarrow n^2 \cdot \log n \in O(n^3) \\ &\Rightarrow n^2 \cdot \log n \in O(n^3) \cap \Omega(n^2) \quad \square \end{aligned}$$

### 2.2 Übungsaufgabe 2.2

[ | 4 ]

Ordnen Sie die folgenden Funktionen nach ihrem Wachstumsgrad in aufsteigender Reihenfolge, d.h. folgt eine Funktion  $g(n)$  einer Funktion  $f(n)$ , so soll  $f(n) \in O(g(n))$  gelten.

$$n, \log n, n^2, n^{\frac{1}{2}}, \sqrt{n^3}, 2^n, \ln n, 1000$$

Mit  $\log$  ist hier der Logarithmus zur Basis 2, mit  $\ln$  der natürliche Logarithmus (Basis  $e$ ) gemeint. Begründen Sie stets Ihre Aussage. Zwei Funktionen  $f(n)$  und  $g(n)$  befinden sich ferner in der selben Äquivalenzklasse, wenn  $f(n) \in \Theta(g(n))$  gilt. Geben Sie an, welche Funktionen sich in derselben Äquivalenzklasse befinden und begründen Sie auch hier ihre Aussage.

Die bearbeitete Menge wird i.F. als  $M_F$  bezeichnet. Die Menge, die gerade alle Elemente von  $M_F$  in aufsteigend sortierter Reihenfolge enthält, wird als  $M'_F$  bezeichnet.

$M_F$  wird mit INSERTSORT in  $M'_F$  hineinsortiert.

Sei  $e \in M_F$ . Für  $e$  wird das Element 1000 gewählt. Da  $|M'_F|$  leer ist, muss 1000 nicht weiter geprüft werden.

$$M'_F = \{1000\}$$

$e$  wird nun über  $M_F$  iteriert, bis  $M'_F = \text{Sorted}(M_F)$ .

$$e = n$$

$$\begin{array}{l} f(n) = n \\ g(n) = 1000 \end{array} \quad \lim_{n \rightarrow \infty} \frac{n}{1000} = \infty \Rightarrow n \notin O(1000)$$

$$\begin{array}{l} f(n) = 1000 \\ g(n) = n \end{array} \quad \lim_{n \rightarrow \infty} \frac{1000}{n} = 0 \Rightarrow 1000 \in O(n) \Rightarrow n \text{ folgt } 1000$$

$$M'_F = \{1000, n\}$$

$$e = \log n$$

$$\begin{array}{l} f(n) = \log(n) \\ g(n) = n \end{array} \quad \lim_{n \rightarrow \infty} \frac{\log(n)}{n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{\ln(2) \cdot n}}{1} = \lim_{n \rightarrow \infty} \frac{1}{\ln(2) \cdot n} = 0 \Rightarrow \log(n) \in O(n) \Rightarrow n \text{ folgt } \log(n)$$

$$\begin{array}{l} f(n) = \log(n) \\ g(n) = 1000 \end{array} \quad \lim_{n \rightarrow \infty} \frac{\log(n)}{1000} = \infty \Rightarrow \log(n) \notin O(1000)$$

$$M'_F = \{1000, \log n, n\}$$

$$e = 4$$

$$\begin{array}{l} f(n) = 4 \\ g(n) = n \end{array} \quad \lim_{n \rightarrow \infty} \frac{4}{n} = 0 \Rightarrow 4 \in O(n) \Rightarrow n \text{ folgt } 4$$

$$\begin{array}{l} f(n) = 4 \\ g(n) = \log(n) \end{array} \quad \lim_{n \rightarrow \infty} \frac{4}{\log(n)} = 0 \Rightarrow 4 \in O(\log(n)) \Rightarrow \log(n) \text{ folgt } 4$$

$$\begin{array}{l} f(n) = 4 \\ g(n) = 1000 \end{array} \quad \lim_{n \rightarrow \infty} \frac{4}{1000} = 0,004 \Rightarrow 4 \in \Theta(1000) \Rightarrow 4 \text{ und } 1000 \text{ befinden sich in der selben } \ddot{\text{A}}\text{-klasse}$$

$$M'_F = \{4, 1000, \log n, n\}$$

$$e = n^2$$

$$\begin{array}{l} f(n) = n^2 \\ g(n) = n \end{array} \quad \lim_{n \rightarrow \infty} \frac{n^2}{n} = \infty \Rightarrow n^2 \notin O(n)$$

$$\begin{array}{l} f(n) = n \\ g(n) = n^2 \end{array} \quad \lim_{n \rightarrow \infty} \frac{n}{n^2} = 0 \Rightarrow n \in O(n^2) \Rightarrow n^2 \text{ folgt } n$$

$$M'_F = \{4, 1000, \log n, n, n^2\}$$

$$e = n^{\frac{1}{2}}$$

$$f(n) = n^{\frac{1}{2}}$$

$$g(n) = n^2 \quad \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{2}}}{n^2} = \lim_{n \rightarrow \infty} \frac{1}{n^{\frac{3}{2}}} = 0 \quad \Rightarrow n^{\frac{1}{2}} \in O(n^2) \quad \Rightarrow n^2 \text{ folgt } n^{\frac{1}{2}}$$

$$f(n) = n^{\frac{1}{2}}$$

$$g(n) = n \quad \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{2}}}{n} = \lim_{n \rightarrow \infty} \frac{1}{n^{\frac{1}{2}}} = 0 \quad \Rightarrow n^{\frac{1}{2}} \in O(n) \quad \Rightarrow n \text{ folgt } n^{\frac{1}{2}}$$

$$f(n) = n^{\frac{1}{2}}$$

$$g(n) = \log(n) \quad \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{2}}}{\log(n)} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{2} \cdot n^{-\frac{1}{2}}}{\frac{1}{n(2) \cdot n}} = \lim_{n \rightarrow \infty} \frac{\ln(2) \cdot n^{\frac{1}{2}}}{2} = \infty \quad \Rightarrow n^{\frac{1}{2}} \notin O(\log(n))$$

$$M'_F = \{4, 1000, \log n, n^{\frac{1}{2}}, n, n^2\}$$

$$e = \sqrt{n}^3$$

$$f(n) = \sqrt{n}^3$$

$$g(n) = n^2 \quad \lim_{n \rightarrow \infty} \frac{\sqrt{n}^3}{n^2} = \lim_{n \rightarrow \infty} \frac{1}{n^{\frac{1}{2}}} = 0 \quad \Rightarrow \sqrt{n}^3 \in O(n^2) \quad \Rightarrow n^2 \text{ folgt } \sqrt{n}^3$$

$$f(n) = \sqrt{n}^3$$

$$g(n) = n \quad \lim_{n \rightarrow \infty} \frac{\sqrt{n}^3}{n} = \lim_{n \rightarrow \infty} n^{\frac{1}{2}} = \infty \quad \Rightarrow \sqrt{n}^3 \notin O(n)$$

$$M'_F = \{4, 1000, \log n, n^{\frac{1}{2}}, n, \sqrt{n}^3, n^2\}$$

$$e = 2^n$$

$$f(n) = 2^n$$

$$g(n) = n^2 \quad \lim_{n \rightarrow \infty} \frac{2^n}{n^2} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\ln(2) \cdot 2^n}{2 \cdot n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\ln^2(2) \cdot 2^n}{2} = \infty \quad \Rightarrow 2^n \notin O(n^2)$$

$$f(n) = n^2$$

$$g(n) = 2^n \quad \lim_{n \rightarrow \infty} \frac{n^2}{2^n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{2 \cdot n}{\ln(2) \cdot 2^n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{2}{\ln^2(2) \cdot 2^n} = 0 \quad \Rightarrow n^2 \in O(2^n) \quad \Rightarrow n^2 \text{ folgt } 2^n$$

$$M'_F = \{4, 1000, \log n, n^{\frac{1}{2}}, n, \sqrt{n}^3, n^2, 2^n\}$$

$$e = \ln n$$

$$f(n) = \ln(n)$$

$$g(n) = 2^n \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{2^n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{\ln(2) \cdot 2^n} = \lim_{n \rightarrow \infty} \frac{1}{n \cdot \ln(2) \cdot 2^n} = 0 \quad \Rightarrow \ln(n) \in O(2^n) \quad \Rightarrow 2^n \text{ folgt } \ln(n)$$

$$f(n) = \ln(n)$$

$$g(n) = n^2 \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{n^2} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{2 \cdot n} = \lim_{n \rightarrow \infty} \frac{1}{2 \cdot n^2} = 0 \quad \Rightarrow \ln(n) \in O(n^2) \quad \Rightarrow n^2 \text{ folgt } \ln(n)$$

$$f(n) = \ln(n)$$

$$g(n) = \sqrt{n}^3 \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{\sqrt{n}^3} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{\frac{3}{2} \sqrt{n}} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{1}{6 \cdot n^{\frac{3}{2}}} = 0 \quad \Rightarrow \ln(n) \in O(\sqrt{n}^3) \quad \Rightarrow \sqrt{n}^3 \text{ folgt } \ln(n)$$

$$f(n) = \ln(n)$$

$$g(n) = n \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{n} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{1} = \lim_{n \rightarrow \infty} \frac{1}{n} = 0 \quad \Rightarrow \ln(n) \in O(n) \quad \Rightarrow n \text{ folgt } \ln(n)$$

$$f(n) = \ln(n)$$

$$g(n) = n^{\frac{1}{2}} \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{n^{\frac{1}{2}}} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{\frac{1}{2} \cdot n^{\frac{1}{2}}} = \lim_{n \rightarrow \infty} \frac{2}{n^{\frac{3}{2}}} = 0 \quad \Rightarrow \ln(n) \in O(n^{\frac{1}{2}}) \quad \Rightarrow n^{\frac{1}{2}} \text{ folgt } \ln(n)$$

$$f(n) = \ln(n)$$

$$g(n) = \log(n) \quad \lim_{n \rightarrow \infty} \frac{\ln(n)}{\log(n)} \stackrel{\text{L'Hospital}}{=} \lim_{n \rightarrow \infty} \frac{\frac{1}{n}}{\frac{1}{\ln(2) \cdot n}} = \ln(2) \quad \Rightarrow \ln(n) \in \Theta(\log(n)) \quad \Rightarrow \ln(n) \text{ und } \log(n) \text{ befinden sich in der selben } \ddot{\text{A}}\text{-klasse}$$

$$M'_F = \{4, 1000, \ln n, \log n, n^{\frac{1}{2}}, n, \sqrt{n}^3, n^2, 2^n\}$$

In der selben Äquivalenzklasse befinden sich zum einen 4 und 1000 und zum anderen  $\log(n)$  und  $\ln(n)$ . Die restlichen Werte sind jeweils alleine in ihrer Äquivalenzklasse.

## 2.3 Übungsaufgabe 2.3

[ | 2 ]

Beweisen oder widerlegen Sie:

$$f(n), g(n) \in O(h(n)) \Rightarrow f(n) \cdot g(n) \in O((h(n))^2)$$

Für diesen Beweis wird der Beweis des dritten Satzes der Summen- und Produkteigenschaften der O-Notation<sup>1</sup> zu Hilfe genommen:

*Beweis.* Sei  $f \in O(h_1)$  und  $g \in O(h_2)$ , dann gibt es ein  $c, n_0$ , so dass  $f(n) \leq c \cdot h_1(n) \forall n \geq n_0$  und ebenso  $c', n'_0$ , so dass  $g(n') \leq c' \cdot h_2(n') \forall n' \geq n'_0$ . Daraus folgt  $f(n'') \cdot g(n'') \leq c \cdot c' \cdot h_1(n'') \cdot h_2(n'') \forall n'' \geq \max(n_0, n'_0)$ , also  $f \cdot g \in O(h_1 \cdot h_2)$ .  $\square$

Setzt man nun  $h_1, h_2 = h$  folgt daraus für den letzten Ausdruck des Beweises  $f(n) \cdot g(n) \in O(h(n) \cdot h(n)) \Rightarrow f(n) \cdot g(n) \in O((h(n))^2)$ .

<sup>1</sup>vgl. Vorlesung, Foliensatz 1 (14.10.), S.33

## 2.4 Übungsaufgabe 2.4

[ | 8 ]

Seien

1.

$$T(n) := \begin{cases} 0, & \text{für } n = 0 \\ 3 \cdot T(n-1) + 2, & \text{sonst} \end{cases}$$

2.

$$S(n) := \begin{cases} c, & \text{für } n = 1 \\ 16 \cdot S(\frac{n}{4}) + n^2, & \text{sonst} \end{cases}$$

Rekurrenzgleichungen ( $c$  ist dabei eine Konstante).

Bestimmen Sie wie in der Vorlesung jeweils die Größenordnung der Funktion  $T : \mathbb{N} \rightarrow \mathbb{N}$  einmalls mittels der (a) Substitutionsmethode und einmal mittels des (b) Mastertheorems. Ihre Ergebnisse sollten zumindest hinsichtlich der O-Notation gleich sein, so dass Sie etwaige Rechenfehler entdecken können! Führen Sie bei (a) auch den Induktionsbeweis, der in der Vorlesung übersprungen wurde!

1. a)

$$\begin{aligned} T(n) &= 3 \cdot T(n-1) + 2 \\ &= 3 * (3 * T(n-2) + 2) + 2 = 3^2 * T(n-2) + 3^2 - 1 \\ &= 3^2 * (3 * T(n-3) + 2) + 8 = 3^3 * T(n-3) + 3^3 - 1 \\ &= \dots \\ &= 3^k * T(n-k) + 3^k - 1 \end{aligned}$$

Wir kommen auf eine sinnvolle Verallgemeinerung der Formel.

Beweis der Formel durch vollständige Induktion:

Induktionsanfang:  $T(0)$  gilt nach Definition.

Induktionsschritt: Sei  $n \in \mathbb{N}$  (s. Aufgabenstellung). Wir nehmen an, dass  $T(n)$  gilt (Induktionsannahme) und zeigen  $T(n+1)$ . Es gilt

$$\begin{aligned} T(n) &= 3 * T(n-1) + 2 \\ T(n+1) &= 3 * T(n+1-1) + 2 \\ &= 3 * T(n) + 2 \end{aligned}$$

$$\begin{aligned} T(n) &= 3^k * T(n-k) + 3^k - 1 \\ T(n+1) &= 3^k * T(n+1-k) + 3^k - 1 \end{aligned}$$

Das zeigt  $T(n+1)$ .

Damit sind der Induktionsanfang und der Induktionsschritt bewiesen. Es folgt, dass  $T(n)$  für alle  $n \in \mathbb{N}$  gilt.

Da die Rekursion bei  $T(0) = 0$ , also  $n-k = 0$  abbricht, wird mit  $k = n$  weiter gerechnet.

$$\begin{aligned} T(n) &= 3^k * T(n-k) + 3^k - 1 \\ &= 3^n * T(n-n) + 3^n - 1 \\ &= 3^n * T(0) + 3^n - 1 \\ &= 3^n * 0 + 3^n - 1 \\ &= 3^n - 1 \in \Theta(3^n) \end{aligned}$$



b) Das Mastertheorem ist auf Aufgabe 1. nicht anwendbar, da die Form

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

bei

$$T(n) := \begin{cases} 0, & \text{für } n = 0 \\ 3 \cdot T(n-1) + 2, & \text{sonst} \end{cases}$$

nicht eingehalten wurde.

2. a)

$$\begin{aligned} S(n) &= 16 \cdot S(\frac{n}{4}) + n^2 \\ &= 16 \cdot S(\frac{16 \cdot S(\frac{n}{4}) + n^2}{4}) + n^2 \\ &= 16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{n}{4}) + n^2}{4}) + n^2}{4}) + n^2 \\ &= 16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{n}{4}) + n^2}{4}) + n^2}{4}) + n^2}{4}) + n^2 \\ &= 16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{16 \cdot S(\frac{n}{4}) + n^2}{4}) + n^2}{4}) + n^2}{4}) + n^2}{4}) + n^2 \end{aligned}$$

Keine sinnvolle Vereinfachung erkennbar. => Substitutionsmethode nicht anwendbar.

b) Die Form

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

ist bei

$$S(n) := \begin{cases} c, & \text{für } n = 1 \\ 16 \cdot S(\frac{n}{4}) + n^2, & \text{sonst} \end{cases}$$

eingehalten. Das Mastertheorem ist daher anwendbar.

I.  $S(n) \in \Theta(n^{\log_b(a)})$ , falls  $f(n) \in O(n^{\log_b(a)-\epsilon})$  für ein  $\epsilon > 0$ .

$$\begin{aligned} f(n) &\in O(n^{\log_b(a)-\epsilon}) \\ n^2 &\in O(n^{\log_4(16)-\epsilon}) \\ n^2 &\in O(n^{2-\epsilon}) \end{aligned}$$

Hierfür kann kein  $\epsilon$  gefunden werden. Daher gilt diese Aussage nicht

II.  $S(n) \in \Theta(n^{\log_b(a)} \cdot \log_2(n))$ , falls  $f(n) \in \Theta(n^{\log_b(a)})$ .

$$\begin{aligned} f(n) &\in O(n^{\log_b(a)}) \\ n^2 &\in O(n^{\log_4(16)}) \\ n^2 &\in O(n^2) \end{aligned}$$

Dies stimmt, daher gilt diese Aussage.

III.  $S(n) \in \Theta(f(n))$ , falls  $f(n) \in \Omega(n^{\log_b(a)+\epsilon})$  für ein  $\epsilon > 0$  **und**  $a \cdot f(\frac{n}{b}) \leq \delta \cdot f(n)$  für ein  $\delta < 1$  und große  $n$ .

$$\begin{array}{ll} f(n) & \in \Omega(n^{\log_b(a)+\epsilon}) \\ n^2 & \in \Omega(n^{\log_4(16)+\epsilon}) \\ n^2 & \in \Omega(n^{2+\epsilon}) \end{array}$$

Dies stimmt für alle  $\epsilon \geq 0$ , also auch für mindestens ein  $\epsilon > 0$ .

$$\begin{array}{ll} a \cdot f(\frac{n}{b}) \leq \delta \cdot f(n) & \backslash \text{einsetzen} \\ 16 \cdot (\frac{n}{4})^2 \leq \delta \cdot n^2 & \backslash \sqrt{()} \\ 4 \cdot \frac{n}{4} \leq \sqrt{\delta} \cdot n & \\ n \leq \sqrt{\delta} \cdot n & \backslash :n \text{ (n ist immer positiv, da } n \in \mathbb{N}, \text{ s. Aufgabenstellung)} \\ 1 \leq \sqrt{\delta} & \backslash ()^2 \\ 1 \leq \delta & \end{array}$$

Damit ist  $\delta \geq 1$  und nicht, wie benötigt,  $\delta < 1$ . Daher gilt diese Aussage nicht.

Da nur II. gilt, gilt  $S(n) \in \Theta(n^{\log_b(a)} \cdot \log_2(n))$ , also  $S(n) \in \Theta(n^2 \cdot \log_2(n))$ .

### 3 Zettel vom 29. 10. – Abgabe: 09. 11.

#### 3.1 Übungsaufgabe 3.1

[ | 3 ]

Gegeben seien die folgenden Code-Fragmente. Geben Sie eine möglichst dichte asymptotische obere Schranke für die Laufzeit der einzelnen Code-Fragmente jeweils in Abhängigkeit von  $n$  an. Begründen Sie Ihre Behauptung. (Es geht hier nicht um die Bedeutung des Codes, nur um die Laufzeit. An der Stelle von  $sum = sum + j$  könnte sinnvoller(er) Code stehen. Die Einrückung gibt den Skopus der Schleifenkonstrukte an.)

ALGO1()	ALGO2()	ALGO3()
1 <b>for</b> $i = 0$ <b>to</b> $n$	1 $i = 1$	1 $i = 1$
2 <b>for</b> $j = n$ <b>downto</b> 1	2 <b>while</b> $i < 2 \cdot n$	2 <b>while</b> $i \cdot i < n$
3 $sum = sum + j$	3 <b>for</b> $j = 1$ <b>to</b> $i$	3 $i = i + 1$
4 <b>for</b> $j = 1$ <b>to</b> $n$	4 $sum = sum + j$	4 $j = n$
5 $sum = sum + j$	5 $i = i + 2$	5 <b>while</b> $j > 1$
		6 $sum = sum + j$
		7 $j = j/2$

1. Der Algorithmus läuft in  $O(n^2)$ .  
Bei beiden inneren **for**-Blöcke laufen jeweils  $n - 1$  mal durch, sodass der innere Block insgesamt  $2 \cdot (n - 1) = 2n - 2$  mal ausgeführt wird. Die äußere **for**-Schleife läuft gerade  $n$  mal; also wird der Block der Länge  $2n - 2$   $n$  mal ausgeführt. Dies führt zu einer Laufzeit von  $n \cdot (2n - 2) = 2n^2 - 2n \in O(n^2)$ .
2. Hauke sagt  $n^2$
3. Hauke sagt was von  $\log(n)$

#### 3.2 Übungsaufgabe 3.2

[ | 2 ]

#### 3.3 Übungsaufgabe 3.3

[ | 3 ]

#### 3.4 Übungsaufgabe 3.4

[ | 8 ]

