Lösungen von Übungsblatt 2
ALP III: Algorithmen, Datenstrukturen und Datenabstraktion
(K. Kriegel)

Tutorium: Wellner, David; Mittwoch, 12:00 - 14:00
Tutorium von Boyan: Berendsohn, Benjamin Aram; Dienstag, 16:00 - 18:00

Boyan Hristov und Julian Habib

2. November 2016

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Σ
/5	/5	/5	/5	

Aufgabe 1

a)

$$f(n) \in O(g(n)) \Rightarrow \exists c_1 > 0 \exists n_{01} \forall n \geq n_{01} : f(n) \leq c_1.g(n) \tag{1}$$
$$h(n) \in \Omega(g(n)) \Rightarrow \exists c_2 > 0 \exists n_{02} \forall n \geq n_{02} : c_2.g(n) \leq h(n) \tag{2}$$

Aus (1) und (2) und $c_1, c_2, f(n), g(n), h(n) \geq 0$ (Da es hier um Laufzeitsanalyse geht)
$\Rightarrow \exists c_1, c_2 \forall n_0 = max(n_{01}, n_{02}) \geq n : f(n).c_2.g(n) \leq c_1.g(n).h(n)$
$\Rightarrow f(n).c_2 \leq h(n).c_1 \Rightarrow f(n)\dfrac{c_2}{c_1} \leq h(n)$

$\Rightarrow \exists c = \dfrac{c_2}{c_1} > 0 \exists n_0 \forall n \geq n_0 : f(n) \leq c.h(n) \Rightarrow f(n) \in O(h(n))$

□

b)

$$f_1(n) \in O(g_1(n)) \Rightarrow \exists c_1 > 0 \exists n_{01} \forall n \geq n_{01} : f_1(n) \leq c_1.g_1(n) \tag{1}$$
$$f_2(n) \in O(g_2(n)) \Rightarrow \exists c_2 > 0 \exists n_{02} \forall n \geq n_{02} : f_2(n) \leq c_2.g_2(n) \tag{2}$$

Aus (1) und (2) $\Rightarrow$
$\exists c_1, c_2 > 0 \exists n_0 = max(n_{01}, n_{02}) \forall n \geq n_0 : f_1(n) \leq c_1.g_1(n) \wedge f_2(n) \leq c_2.g_2(n)$
$\Rightarrow f_1(n) + f_2(n) \leq c_1.g_1(n) + c_2.g_n(n) \leq max(g_1(n), g_2(n)).c_1 + max(g_1(n), g_2(n)).c_2 =$
$= max(g_1(n), g_2(n))(c_1 + c_2)$

$\Rightarrow \exists c = c_1 + c_2 > 0 \exists n_0 \forall n \geq n_0 : f_1(n) + f_2(n) \leq max(g_1(n), g_2(n)).c$
$\Rightarrow f_1(n) + f_2(n) \in O(max(g_1(n), g_2(n)))$

□

Aufgabe 2

a)

$$\sqrt{n}(log(n))^2 < \sqrt{n}\sqrt[2]{n} = \qquad \text{(Nach 9. Satz von "Werkzeuge und Grundlagenäus der Vorlesung)}$$
$$= \sqrt{n}^2 = n$$
$$\Rightarrow \sqrt{n}(log(n))^2 \in o(n) \ \wedge \ \sqrt{n}(log(n))^2 \in O(n)$$

b)

$$\frac{n^2}{log\ \ n} > \frac{n^2}{n} = n \qquad \text{(für } n \geq 2)$$
$$\Rightarrow \frac{n^2}{log\ \ n} \in \omega(n) \ \wedge \ \frac{n^2}{log\ \ n} \in \Omega(n)$$

c)

$$4^n = (2^2)^n = 2^{2n} = 2^n.2^n > 2^n \Rightarrow 4^n \notin O(2^n)$$

Aufgabe 3

a)

$$f_1(n) = log((n!)^2) = 2.log(n!) \quad (1) \qquad \text{(Grund: } log_a(n^b) = b.log_a n)$$
$$log(n!) \in \Theta(n.log(n)) \quad (2) \qquad \text{(5. Regel aus "Grundlagen und Werkzeugenïn der Vorlesung)}$$

$$(1) \wedge (2) \Rightarrow f_1(n) = log((n!)^2) \in \Theta(n.log(n)) \text{ ,da 2 eine Konstante ist}$$

b)

$$f_2(n) = log((n^2)!)$$
$$\text{Sei } x = n^2 \Rightarrow \qquad \text{(Nach 5. Regel aus der Vorlesung)}$$
$$\exists c_1, c_2 > 0 : c_1.x.log(x) \leq log(x!) \leq c_2.x.log(x) \Rightarrow \qquad \text{(Wider x einsetzen)}$$
$$\Rightarrow \exists c_1, c_2 > 0 : c_1.n^2.log(n^2) \leq log((n^2)!) \leq c_1.n^2.log(n^2)$$
$$\Rightarrow \exists c_1, c_2 > 0 : c_1.n^2.2.log(n) \leq log((n^2)!) \leq c_1.n^2.2.log(n)$$

$$\Rightarrow f_2(n) = log((n^2)) \in \Theta(n^2.log(n)) \qquad \text{(Da } c_1.2,\ c_2.2 \text{ Konstanten sind)}$$

c)

$$f_3(n) = log(log(n!)) \Rightarrow \text{Nach 5. Regel}$$
$$\exists c_1, c_2 > 0 : log(c_1.n.log(n)) \leq log(log(n!)) \leq log(c_2.n.log(n)) \Rightarrow \qquad \text{(Nach log(a.b) = log(a) + log(b))}$$
$$\Rightarrow log(c_1) + log(n) + log(log(n) \leq log(log(n!)) \leq log(c_2) + log(n) + log(log(n))$$

$$\Rightarrow \exists c_1, c_2 > 0 : c_1.log(n) \leq log(log(n!)) \leq c_2.log(n) \Rightarrow \qquad \text{(Nach Def. von } \Theta)$$
$$\Rightarrow log(log(n!)) \in \Theta(log(n))$$

d)

$$f_4(n) = log(n^n) = n.log(n) \qquad \text{(da } log_a(n^b) = b.log_a n)$$
$$\Rightarrow f_4(n) \in \Theta(n.log(n))$$

e)

$$f_5(n) = log(4^{n!}) = n!.log(4) \qquad \text{(da } log_a(n^b) = b.log_a n)$$
$$= 2n! \in \Theta(n!)$$

Aufgabe 4

a)

3 + 4 + 2*(4 - 2)

2*(4 - 2) ≡ 2 4 2 - *
4 + 2*(4-2) ≡ 4 2 4 2 - * +

3 + 4 + 2*(4 - 2) ≡ 3 4 2 4 2 - * + +

2 + (3 * 3 - 2 * 4)

3 * 3 ≡ 3 3 *
2 * 4 ≡ 2 4 *
3 * 3 - 2 * 4 ≡ 3 3 * 2 4 * -

2 + (3 * 3 - 2 * 4) ≡ 2 3 3 * 2 4 * - +

b)

In dieser Aufgabe haben wir den Stack als ein Array (also so []) vorgestellt. Alle Schritte, in denen nur weitere Operanden geholt werden haben wir übersprüngen.

3 4 2 4 2 - * + +

$[3, 4, 2, 4, 2, -] \rightarrow [3, 4, 2, 2] \rightarrow [3, 4, 2, 2, *] \rightarrow [3, 4, 4] \rightarrow [3, 4, 4, +] \rightarrow [3, 8] \rightarrow [3, 8, +] \rightarrow [11] = 1$

2 3 3 * 2 4 * - +

$[2, 3, 3, *] \rightarrow [2, 9] \rightarrow [2, 9, 2, 4, *] \rightarrow [2, 9, 8] \rightarrow [2, 9, 8, -] \rightarrow [2, 1] \rightarrow [2, 1, +] \rightarrow [3] = 3$