

# Nachbesprechung Serie 2

15.3.2018



# Allgemein

- Serie wurde gut gelöst
- Alte Lösungen bitte nicht abgeben, ab jetzt nicht mehr korrigiert
- Schöner schreiben...

# 1.a

- Laufzeit [...] **mindestens**  $O(n)$ ", keinen Sinn macht
- Wir vergleichen ein **Minimum** mit einer **Obergrenze**

# 1.b

Definitionen im Buch Kapitel 3 (Seite 46-52)

Kombination aus Ober- und Unterschranke ergibt O-Notation

## 2.b

Um zu zeigen, dass  $\Theta(\log_a n) = \Theta(\log_b n)$ , müssen wir zeigen, dass es eine Konstante  $c$  gibt, so dass  $\log_a n = c * \log_b n$ . Wir finden:

$$\log_a n = c * \log_b n$$

$$\frac{\log_b n}{\log_b a} = c * \log_b n$$

$$\frac{1}{\log_b a} = c$$

## 2.c

Ähnlich wie 2.b:

Zeige: es gibt eine Konstante  $c$ , so dass  $a^n = c * b^n$

$\Rightarrow (a/b)^n = c \Rightarrow a/b$  ist kleiner  $< 1$ , d.h.  $\lim_{n \rightarrow \infty} (a/b)^n = 0 \Rightarrow$  es gibt kein  $c$

$\Rightarrow \Theta(a^n) \neq \Theta(b^n)$

# 3

Detaillierte Lösung auf Ilias

# 6 lösbare Problemgrösse

Berechnen Sie die lösbare Problemgrösse in der gegebenen Zeit für Algorithmen mit verschiedener Zeitkomplexität, welche in der Tabelle gegeben sind. Nehmen Sie an, jede Operation dauere 0.01s. **(2 Punkte)**

Zeitkomplexität $T(n)$	Problemgrösse lösbar in 10s	Problemgrösse lösbar in 1000s
$10n$		
$2n^3$		
$n^{2.5}$		
$2\log_2(8n)$		
$2^{2n}$		



# 6 lösbare Problemgrösse

Berechnen Sie die lösbare Problemgrösse in der gegebenen Zeit für Algorithmen mit verschiedener Zeitkomplexität, welche in der Tabelle gegeben sind. Nehmen Sie an, jede Operation dauere 0.01s. **(2 Punkte)**

Zeitkomplexität $T(n)$	Problemgrösse lösbar in 10s	Problemgrösse lösbar in 1000s
$10n$	100	10'000
$2n^3$	7.937	36.84
$n^{2.5}$	15.84	100
$2 \log_2(8n)$	$2^{497}$	$2^{49997}$
$2^{2n}$	$\log_2(1000) = 4.98$	$\log_2(100000) = 8.30$

