technische universität
dortmund

SoSe 2017

C. Sohler
A. Krivošija, A. Rey, H. Sandvoß

DAP2 – Präsenzübung 1

Besprechung: 26.04.2017 — 28.04.2017

Abgabe:

Präsenzübungen müssen nicht zu Hause bearbeitet werden, sondern werden unter Anleitung während der Übung erarbeitet.

Präsenzaufgabe 1.1: (Erste Schritte mit der O–Notation)

Gegeben sei die Funktion

$$f(n) = \begin{cases} 2017^n & \text{falls } n \leq 20 \\ n^{2017} & \text{falls } 20 < n \leq 201 \\ 2017n & n > 201. \end{cases}$$

Bestimmen Sie eine Funktion $g(n)$, sodass $f(n) \in \Theta(g(n))$ gilt.

Welchen Einfluss haben die Konstanten 20, 201 und 2017 auf die Wahl Ihrer Funktion gehabt?

Bestimmen Sie auch eine nicht konstante Funktion $h(n)$, sodass $f(n) \in \omega(h(n))$ gilt.

Präsenzaufgabe 1.2: (Anschauung der Landau-Notation)

Geben Sie je ein Beispiel für Funktionen $a(n)$, $b(n)$, $c(n)$ und $d(n)$, sodass die Aussagen

- $a(n) \in O(1)$
- $b(n) \in o(1)$
- $c(n) \in \Omega(1)$
- $d(n) \in \omega(1)$

gelten, und beweisen Sie die Korrektheit der Aussagen für die von Ihnen ausgewählten Funktionen.

Was bedeuten diese Aussagen umgangssprachlich?

Präsenzaufgabe 1.3: (Asymptotische Komplexität)

Geben Sie *jeweils* eine möglichst kleine Funktion $g : \mathbb{N} \to \mathbb{R}^+$ an, für die gilt:

(a) $2^n/50 - 12n^{10} + n^9 = O(g(n))$

(b) $5n + 2\sqrt{n}\ln n = O(g(n))$

und eine möglichst große Funktion $h : \mathbb{N} \to \mathbb{R}^+$, für die gilt:

(c) $n^4 - 10n^3 + 5 = \Omega(h(n))$

(d) $n \log n^2 - 2\sqrt{n}\ln^2 n = \Omega(h(n))$

Beweisen Sie Aussagen (a) – (d) für die von Ihnen gegebenen Funktionen gemäß der Definitionen der Landau-Symbole aus der Vorlesung.

Hinweis: Bei dieser Aufgabe dürfen Sie die Aussagen nutzen, die in der Vorlesung bewiesen wurden.

Präsenzaufgabe 1.4: (Laufzeitanalyse: Primfaktoren)

Führen Sie eine Worst-Case Laufzeitanalyse für den folgenden Algorithmus durch, der bei Eingabe einer ganzen Zahl n alle ihre Primfaktoren im Array A zurückgibt.

```
Primfaktor(int n):
j ← 1
m ← ⌊√n⌋
i ← 2
while i ≤ m do
    if n teilbar durch i then
        A[j] ← i
        n ← n/i
        j ← j + 1
    else
        i ← i + 1
if n > 1 then
    A[j] ← n
return A
```

Ordnen Sie die Laufzeit $f(n)$ bei Eingabe der Zahl n in die O-Notation ein, d.h. finden Sie eine möglichst kleine Funktion $g(n)$, sodass $f(n) \in O(g(n))$ ist.