Universität Klagenfurt
Informatik – Systemsicherheit
M. Morak · R. Wigoutschnigg

UE Algorithmen und Datenstrukturen
SS 2019
Übungstermine: siehe ZEUS

Übungsblatt 1

Aufgabe 1.1: Komplexitätsklassen
Es seien $\log(n) := \log_2(n)$, $\ln(n) := \log_e(n)$ und $\varepsilon \in \mathbb{R}$ eine Konstante mit $0 < \varepsilon < 1$. Ordnen Sie die folgenden Komplexitätsklassen nach ihrer Größe und kennzeichnen Sie (gegebenenfalls) gleichmächtige Komplexitätsordnungen.

$O(1)$,	$O(0,2 \cdot n)$,	$O(n^3)$,	$O(2^n)$,
$O(n \cdot \log(n))$,	$O(n^n)$,	$O(n^{\log(7)})$,	$O(n \cdot \log(n) \cdot \log(\log(n)))$,
$O(\ln(n))$,	$O(\log(n^2))$,	$O(n^2)$,	$O(5 \cdot n)$,
$O(n^{1+\varepsilon})$,	$O(n \cdot (\log(n))^{1+\varepsilon})$,	$O(n \cdot \ln(n))$,	$O((\log(n))^2)$.

Aufgabe 1.2: Landau-Symbole

a) Es bezeichne $[x]$ den ganzzahligen Anteil der reellen Zahl x. Für welche Paare von Funktionen f_i und g_i $(1 \leq i \leq 5)$ gilt $f_i \in O(g_i)$, $f_i \in \Omega(g_i)$ bzw. $f_i \in \Theta(g_i)$?

$f_1(n) = [\sqrt{n}]$	$g_1(n) = 1000 \cdot n$
$f_2(n) = [\sqrt[3]{n}]$	$g_2(n) = [\sqrt{n}]$
$f_3(n) = n^2$	$g_3(n) = [n \cdot \log(n)]$
$f_4(n) = 198 \cdot n^2 - 12 \cdot n + 55$	$g_4(n) = n^2$
$f_5(n) = [n^2 \cdot \sqrt[5]{n}]$	$g_5(n) = [n^2 \cdot \sqrt{(\log(n))^{17,5}}]$

b) Existieren Funktionen $f, g : \mathbb{N} \to \mathbb{N} + 1$ mit $f \notin O(g)$ und $g \notin O(f)$?
Beweisen Sie Ihre Aussage!

Aufgabe 1.3: Komplexität
Gegeben ist folgender Algorithmus zum Sortieren einer Liste L_1 der Länge n:

1. Lege eine leere Liste L_2 an.
2. Verschiebe das größte Element der Liste L_1 und an den Anfang der Liste L_2.
3. Solange die Liste L_1 nicht leer ist, führe Schritt 2 aus.
4. Gib die Liste L_2 aus.

a) Bestimmen Sie die asymptotische Ordnung Θ des Algorithmus im best und im worst case.
b) Ist der Algorithmus schneller, wenn die Liste L_1 bereits vorsortiert ist? Begründen Sie Ihre Antwort.

Aufgabe 1.4: Perfekte Zahlen
Recherchieren Sie den Begriff *Perfekte Zahlen* und geben Sie einen Algorithmus an, der überprüft ob eine beliebige Zahl n eine Perfekte Zahl ist. Weiters geben Sie die Komplexitätsklasse Ihres Algorithmus exakt an (Θ). Sollten dies nicht möglich sein, so geben Sie möglichst scharfe Schranken (O, Ω) an. Wie verhält sich die Sachlage bei *Armstrong Zahlen*?
Hinweis: Recherchieren Sie, wie viele Armstrong Zahlen es gibt.

Bitte wenden!

Aufgabe 1.5: Matritzenmultiplikation

Es seien M_1 eine 10×20 Matrix, M_2 eine 20×50 Matrix, M_3 eine 50×1 Matrix und M_4 eine 1×100 Matrix mit ganzzahligen Werten.

a) Wie viele Möglichkeiten (Reihenfolgen der erforderlichen Matrizenmultiplikationen) existieren, um das Matrizenprodukt $M = M_1 \cdot M_2 \cdot M_3 \cdot M_4$ zu ermitteln.
Hinweis: Die Multiplikation von Matrizen ist assoziativ.

b) Wie viele Zahlen-Multiplikationen werden (minimal und maximal) zur Berechnung von M benötigt?