Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement Informatik
Markus Püschel David Steurer Peter Widmayer
Chih-Hung Liu

23. Oktober 2017

Algorithmen & Datenstrukturen Blatt 5 HS 17

Übungsstunde (Raum & Zeit): ______________________

Abgegeben von: ______________________

Korrigiert von: ______________________

erreichte Punkte: ____

Aufgabe 5.1 *Eigenschaften der $\mathcal{O}$-Notation.*

Beweisen Sie oder widerlegen Sie mit einem Gegenbeispiel die folgenden Aussagen. Es seien $f, g : \mathbb{N} \to \mathbb{R}^+$.

a) $n^2 \geq \Omega(n)$

b) $2^n \leq \mathcal{O}(n^{10})$

c) $\ln^{1000} n \geq \Omega(\sqrt{n})$

d) $27^{\log_9 n} = \Theta(n \cdot \sqrt{n})$

e) Es gilt $f \geq \Omega(g)$ genau dann, wenn $g \leq \mathcal{O}(f)$.

f) Seien $f_1, f_2 \geq \Omega(g)$, $f_1(n) > f_2(n)$ für $n \geq 1$ und $f(n) := f_1(n) - f_2(n)$. Dann ist auch $f \geq \Omega(g)$.

g) Seien $f_1, f_2 \geq \Omega(g)$ und $f(n) := \frac{f_1(n)}{f_2(n)}$. Dann ist auch $f \geq \Omega(g)$.

h) Für alle $a, b \in \mathbb{N}$, $a \leq b$, gilt $n^{1/a} = \Theta(n^{1/b})$.

Aufgabe 5.2 *Relationen der Komplexität.*

Seien $f, g : \mathbb{N} \to \mathbb{R}^+$ zwei Funktionen. Im Folgenden betrachten wir die Relationen $f \leq \mathcal{O}(g)$, $f \geq \Omega(g)$ sowie $f = \Theta(g)$. In dieser Aufgabe ist jede Ihrer Antworten zu begründen.

a) Geben Sie für jede der drei zuvor genannten Relationen an, ob sie reflexiv, symmetrisch und transitiv ist.

	$f \leq \mathcal{O}(g)$	$f \geq \Omega(g)$	$f = \Theta(g)$
Reflexiv			
Symmetrisch			
Transitiv			

b) Ist eine der drei Relationen eine Äquivalenzrelation?

c) Geben Sie zwei Funktionen f und g an, sodass weder $f \leq \mathcal{O}(g)$ noch $g \leq \mathcal{O}(f)$ gelten.

Aufgabe 5.3 *Rekursion.*

Für diese Aufgabe dürfen Sie annehmen, dass n eine Zweierpotenz ist, und dass a und c positive Konstanten sind.

a) Gegeben sei eine Funktion $f : \mathbb{N} \to \mathbb{R}^+$ mit $f(1) = c$ und $f(n) = 2 \cdot f(\frac{n}{2}) + a \cdot n$ für $n \geq 2$. Zeigen Sie:

$$f(n) = c \cdot n + a \cdot n \cdot \log_2 n.$$

b) Gegeben sei eine Funktion $f : \mathbb{N} \to \mathbb{R}^+$ mit $f(1) = c$ und $f(n) = f(\frac{n}{2}) + a \cdot n$ für $n \geq 2$. Zeigen Sie:

$$f(n) \leq c + 2 \cdot a \cdot n.$$

Aufgabe 5.4 *Mehrheitsvotum.*

Gegeben sei eine Menge von n Objekten $O_1, \ldots, O_n$. Wir wollen entscheiden, ob diese ein *Mehrheitsvotum* besitzt. Dies ist genau dann der Fall, wenn (echt) mehr als die Hälfte aller Objekte gleich sind. Entwerfen Sie einen Algorithmus für dieses Problem. Die einzig erlaubte Elementaroperation ist der Gleichheitstest, d.h. für zwei Objekte O_i und O_j können Sie in Zeit $\mathcal{O}(1)$ prüfen, ob sie gleich sind. Sie dürfen aber **nicht** annehmen, dass die Einträge aus einem geordneten Universum (z.B. $\mathbb{N}$) stammen, also mittels "$<$" verglichen werden können.

a) Überlegen Sie zunächst, wie ein naiver Algorithmus für das Problem aussehen könnte. Welche Laufzeit hat Ihre Lösung?

b) Geben Sie einen Algorithmus an, der das Problem mittels Divide-and-Conquer in Zeit $\mathcal{O}(n \log n)$ löst. Beweisen Sie auch, dass die Laufzeit Ihres Algorithmus $\leq \mathcal{O}(n \log n)$ ist.

   Hinweis: Sie dürfen annehmen, dass $n = 2^k$ für ein $k \in \mathbb{N}$ gilt. Wenn man die gegebene Menge der Objekte in zwei gleich grosse Teile aufspaltet, wie kann man dann aus den beiden Teillösungen etwas für die Gesamtlösung schlussfolgern?

c) Ist die Lösung aus Aufgabenteil b) bestmöglich, oder können Sie eine noch schnellere angeben?

Bitte beschreiben Sie Ihren Algorithmus anschaulich und/oder in **kommentiertem** Pseudocode, sodass Ihre Idee verständlich wird. Bitte verzichten Sie darauf, Java-Code abzugeben.

Abgabe: Am Montag, den 30. Oktober 2017 zu Beginn Ihrer Übungsgruppe.