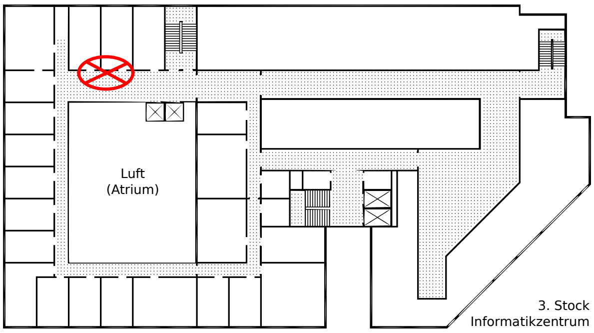
Abteilung Algorithmik **Winter 2013/14**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Stephan Friedrichs

Algorithmen und Datenstrukturen
Übung 3 vom 04. 12. 2013

Abgabe der Lösungen bis zum Mittwoch, den 18. 12. 2013 um 13:00 im Hausaufgabenrückgabeschrank.



Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!

Aufgabe 1 (O-Notation): In der Vorlesung wurde die O-Notation zur Abschätzung der Laufzeit eingeführt.

a) Suche für die folgenden Funktionen jeweils geeignete Konstanten c sowie n_0 und zeige mit Hilfe dieser Konstanten, dass die jeweilige Funktion in der angegebenen Klasse liegt.

$$\begin{aligned}
f_1(n) &= \frac{1}{n^2} \in \mathrm{O}(1) \\
f_2(n) &= 10n^3 + 5n^2 \in \mathrm{O}(n^3) \\
f_3(n) &= 10000n \in \mathrm{O}(n^3) \\
f_4(n) &= \log_2 n \in \mathrm{O}(\log_{10} n)
\end{aligned}$$

b) Kreuze an, in welchen Klassen die jeweilige Funktion liegt (ohne Begründung).

$f(n)$	$\Omega(1)$	$\Theta(1)$	$\mathrm{O}(1)$	$\Omega(n)$	$\Theta(n)$	$\mathrm{O}(n)$	$\Omega(n^2)$	$\Theta(n^2)$	$\mathrm{O}(n^2)$
1337									
$2n + 5\log n$									
1.01^n									
$2n \log n$									
$3.14n$									
$3\log n + 10$									

(8+6 Punkte)

Aufgabe 2 (O-Notation): Beweise Satz 3.12 aus der Vorlesung: Seien $f, g : \mathbb{N} \to \mathbb{R}$ zwei Funktionen. Dann gilt:

a) $f \in \Theta(g) \Leftrightarrow g \in \Theta(f)$

b) $f \in \Theta(g) \Leftrightarrow f \in \mathrm{O}(g)$ und $f \in \Omega(g)$

c) $f \in \mathrm{O}(g) \Leftrightarrow g \in \Omega(f)$

(3+3+3 Punkte)

Aufgabe 3 (Laufzeitkomplexität von Algorithmen): Du hast einen Input von n Elementen und einen Algorithmus, der einzelne Elemente bearbeitet. Das kostet $n^2 + 20 \log n$ Operationen pro Element. Alle Elemente müssen bearbeitet werden.

a) Wie lange dauert das (mit und ohne O-Notation)?

b) Die Bearbeitung zweier Elements ist unabhängig voneinander, man kann sie also parallelisieren. Dazu stehen 4 Kerne zur Verfügung. Wie wirkt sich dies auf die Gesamtlaufzeit aus (mit und ohne O-Notation)?

c) Alternativ kannst du ein Preprocessing verwenden, welches in $17n^2 - 3n + 715$ läuft und danach die Bearbeitungszeit pro Element auf $n \log n$ reduziert, allerdings ist die Bearbeitung der Elemente dann nicht mehr parallelisierbar. Wie wirkt sich das auf die Gesamtlaufzeit aus (mit und ohne O-Notation)?

d) Für welche Möglichkeit (ursprünglicher Algorithmus, b), oder c)) entscheidest du dich, um die Leistungsfähigkeit des Programms für große Inputmengen zu maximieren? Begründe deine Wahl.

(4+4+5+4 Punkte)

Aufgabe 4 (Komplexitätsklassen): Sortiere die folgenden Funktionsklassen nach Inklusion (ohne Begründung). Kennzeichne identische Klassen.

$$\mathrm{O}(n^2 - n),\ \mathrm{O}(\log_2(n^2)),\ \mathrm{O}(3^n),\ \mathrm{O}(n \log_7 15),\ \mathrm{O}(3),\ \mathrm{O}(3n+7),\ \mathrm{O}(2^n),\ \mathrm{O}(1),\ \mathrm{O}(n^2)$$

(Hinweis: Die Mengen $A = \{1,2,3\}$, $B = \{1,2\}$, $C = \{1,2,3,4\}$ und $D = \{1,2,3\}$ nach Inklusion sortiert sind: $B \subseteq A = D \subseteq C$.) **(10 Punkte)**

Aufgabe 5 (Queues): Betrachte die Warteschlange Q (Abbildung 1). Führe nacheinander die folgenden Operationen aus; zeichne dabei nach jeder Operation Q und markiere head[Q] und tail[Q] in der Zeichnung. Bei DEQUEUE-Operationen gib zusätzlich das zurückgegebene Element an.

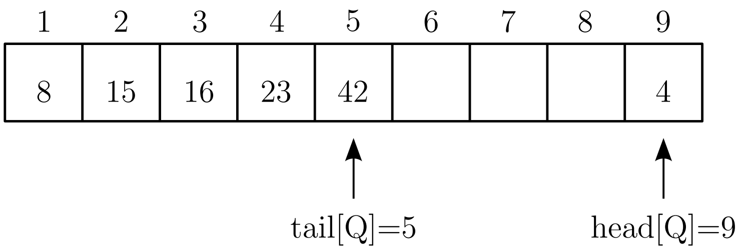


Abbildung 1: Die Warteschlange Q.

a) DEQUEUE(Q)

b) DEQUEUE(Q)

c) ENQUEUE(Q, 7)

d) ENQUEUE(Q, 5)

e) DEQUEUE(Q)

(2+2+2+2+2 Punkte)