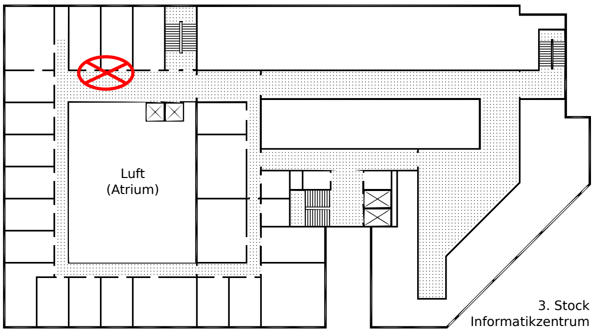
Abteilung Algorithmik **Winter 2015/16**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer
Jan-Marc Reinhardt

Algorithmen und Datenstrukturen
Übung 3 vom 09. 12. 2015

Abgabe der Lösungen bis zum Mittwoch, den 06. 01. 2016 um 11:15 im Hausaufgabenrückgabeschrank.

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



Aufgabe 1 ($\mathcal{O}$-Notation): In der Vorlesung wurde die $\mathcal{O}$-Notation zur Abschätzung der Laufzeit eingeführt.

a) Suche für die folgenden Funktionen jeweils geeignete Konstanten c (bzw. c_1 und c_2) sowie n_0 und zeige mit Hilfe dieser Konstanten, dass die jeweilige Funktion in der angegebenen Klasse liegt.

$$\begin{array}{rcrcl} f_1(n) & = & n - 3\sqrt{n} & \in & \Omega(n) \\ f_2(n) & = & n^2 + n\log_2 n & \in & \Omega(n^2) \\ f_3(n) & = & \log_2(n^3) & \in & \Theta(\log_3 n) \\ f_4(n) & = & 3n^3 + 6n^2 & \in & \mathcal{O}(n^3) \\ f_5(n) & = & \frac{1}{n} + 5 & \in & \mathcal{O}(1) \end{array}$$

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

$f(n)$	$\mathcal{O}(1)$	$\Omega(1)$	$\Theta(1)$	$\mathcal{O}(n)$	$\Omega(n)$	$\Theta(n)$	$\mathcal{O}(n^2)$	$\Omega(n^2)$	$\Theta(n^2)$
$\frac{1.67n^2}{4.93n}$									
$7\log(n^8)n$									
$3^{2\log_3(n)}$									
$5\log(n)\sqrt{n}$									
3^{42}									
$\sqrt{n} - n^2$									

(10+6 Punkte)

Aufgabe 2 ($\mathcal{O}$-Notation): Im Folgenden sollen die Definitionen der $\mathcal{O}$-Notationen aus der Vorlesung verwendet werden. Mit der $\mathcal{O}$-Notation lassen sich Funktionen nach ihrem asymptotischen Wachstum ordnen. Seien $f, g, h : \mathbb{N} \to \mathbb{R}$ Funktionen. Wir definieren folgende Relationen:

$$f \preceq g \Leftrightarrow f \in \mathcal{O}(g)$$
$$f \simeq g \Leftrightarrow f \in \Theta(g)$$

Beweise folgende Aussagen[1]:

a) $f \preceq f$

b) $f \preceq g$ und $g \preceq h \Rightarrow f \preceq h$

c) $f \preceq g$ und $g \preceq f \Rightarrow f \simeq g$

Gilt für b) und c) auch die Umkehrung?

d) Zeige oder widerlege: $f \preceq g$ und $g \preceq h \Leftarrow f \preceq h$

e) Zeige oder widerlege: $f \preceq g$ und $g \preceq f \Leftarrow f \simeq g$

(1+4+4+3+3 Punkte)

Aufgabe 3 (Laufzeitkomplexität von Algorithmen): Du hast einen Input von n Elementen und einen Algorithmus, der einzelne Elemente bearbeitet. Das kostet $n^2 + 7\log n$ Operationen pro Element. Alle Elemente müssen bearbeitet werden.

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

b) Die Bearbeitung zweier Elemente ist unabhängig voneinander, man kann sie also parallelisieren. Dazu stehen 8 Kerne zur Verfügung. Wie wirkt sich dies auf die Gesamtlaufzeit aus (mit und ohne Θ-Notation)?

c) Alternativ kannst du ein Preprocessing verwenden, welches in $3n\log n + 4n - 7$ läuft und danach die Bearbeitungszeit pro Element auf $\log n$ reduziert, allerdings ist die Bearbeitung der Elemente dann nicht mehr parallelisierbar. Wie wirkt sich das auf die Gesamtlaufzeit aus (mit und ohne Θ-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.

$$\Omega(\sqrt{n} + 3n), \Omega(\log_2(n)), \Omega(1), \Omega(\log_{10}(n^3)), \Omega(7), \Omega(3^n), \Omega(\sqrt{n} + 2\log_4(n)),$$
$$\Omega(5n^2 + 2n^3), \Omega(4^n), \Omega(7n^5 + 3n^4 - 2n^2), \Omega(4n - 1), \Omega(n!), \Omega(7n^3)$$

(Hinweis: Die Mengen $A = \{1, 2, 3\}$, $B = \{1, 2\}$, $C = \{1, 2, 3, 4\}$ und $D = \{1, 2, 3\}$ nach Inklusion sortiert sind: $B \subseteq A = D \subseteq C$.) **(12 Punkte)**

[1] Für diejenigen unter euch, die Spaß an Mathematik haben: Wir wollen darauf hinaus, dass $\preceq$ eine Halbordnung auf den durch $\simeq$ gegebenen Äquivalenzklassen ist.