TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Sprachen und Beschreibungsstrukturen
Grundlagen: Algorithmen und Datenstrukturen
Prof. Dr. Helmut Seidl, J. Kranz, A. Reuss, R. Vogler

SS 2016
Übungsblatt 2
24.04.2016

Abgabe: 01.05.2016 (bis 23:59 Uhr)

Aufgabe 2.1 (P) Division

Wir betrachten die Schulmethode der Division. Hierbei untersuchen wir die Zahl der (elementaren) Operationen, die wir benötigen, um eine n-stellige Zahl durch eine einzelne *Ziffer* ganzzahlig mit Rest zu teilen. Für zwei Zahlen $a \in \mathcal{N}_0$ und $b \in \mathcal{N}$ bezeichne $g(a, b) := \lfloor a/b \rfloor$ das Ergebnis der ganzzahligen Division von a durch b, sowie $r(a, b) := a - g(a, b) \cdot b$ den Rest der ganzzahligen Division. Weiterhin bezeichne $a.b$ die Konkatenation zweier Ziffern a und b.

Der folgende Pseudocode beschreibt die Schulmethode. Der Dividend bestehe aus den Ziffern $x_1 \dots x_n$ und der Divisor sei $y > 0$. Der Zähler i ist die aktuell betrachtete Stelle und $z_1.z_2 \dots$ die Ziffern des Ergebnisses der ganzzahligen Division.

Input: Ziffer[] $(x_1, x_2, \dots, x_n)$, Ziffer y
1 int $i := 1$
2 Ziffer $x_0 := 0$ /* Hilfsziffer zur Vermeidung von Fallunterscheidung */
3 **while** $i \leq n$ **do**
4 **if** $x_{i-1} > 0$ **then**
5 Ziffer $z_i := g(x_{i-1}.x_i, y)$
6 $x_i := r(x_{i-1}.x_i, y)$
7 $x_{i-1} := 0$ /* Kann auch weggelassen werden */
8 **end**
9 **else**
10 Ziffer $z_i := g(x_i, y)$
11 $x_i := r(x_i, y)$
12 **end**
13 $i := i + 1$
14 **end**
15 Das Ergebnis der Division ist $z_1.z_2 \dots$, der Rest is x_n.

Algorithm 1: Ganzzahlige Division mit Rest

Beispiel: 36811 ganzzahlig dividiert durch 4. Der Strich markiert die aktuelle Position. Auf der linken Seite sind die Ziffern $x_0 \dots x_n$, auf der rechten Seite die wachsende Ziffernfolge $z_1 \dots z_n$ abgebildet.

$$
\begin{aligned}
036811 &\to 0 \\
\mid & \\
036811 &\to 09 \\
\mid & \\
000811 &\to 092 \\
\mid & \\
000011 &\to 0920 \\
\mid & \\
000011 &\to 09202 \\
\mid & \\
000003 &\to 09202 \text{ Rest } 3 \\
\mid &
\end{aligned}
$$

Für die Auswertung von $g(x_i.x_{i+1}, y)$ und $r(x_i.x_{i+1}, y)$ verwenden wir im Voraus berechnete Tabellen. Wir *legen fest*, dass in unserem Modell jeder Vergleich zweier Zahlen, jede Konkatenation zweier Ziffern, jede Zuweisungsoperation sowie jede Nachschlageoperation in den Tabellen eine *Grundoperation* darstellt.

Wie viele Grundoperationen hat die Schulmethode in unserem Rechenmodell im schlimmsten Fall, wenn man eine Zahl mit n Ziffern ganzzahlig durch eine Ziffer teilt?

Aufgabe 2.2 (P) Induktion

Beweisen Sie folgende Aussagen durch Induktion für alle $n \in \mathbb{N} := \{1, 2, 3, 4, 5, \dots\}$:

(a) $n^{\frac{n}{2}} \leq n! \leq n^n$,

(b) $19 | (5 \cdot 2^{3n+1} + 3^{3n+2})$ (In Worten: 19 teilt $(5 \cdot 2^{3n+1} + 3^{3n+2})$),

Aufgabe 2.3 (P) $\mathcal{O}$-Notation

Kreuzen Sie in den Zeilen (a) bis (f) jeweils das zu den Funktionen stärkste passende Symbol an. Das heißt, wenn $\Delta = o$ (bzw. $\Delta = \Theta$) möglich ist, wählen Sie $\Delta = o$ (bzw. $\Delta = \Theta$) und nicht $\Delta = \mathcal{O}$. Falls die Funktionen unvergleichbar sind, kreuzen Sie „u.“ an.

Bsp.:	n	$\in \Delta(n^2)$	☒ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(a)	$7n^3$	$\in \Delta(3n^7)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(b)	$\sqrt[3]{\sqrt{3n}}$	$\in \Delta\left(\sqrt{\sqrt[3]{2n}}\right)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(c)	$n!$	$\in \Delta(4^n)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(d)	n	$\in \Delta((2+(-1)^n)n)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(e)	$\sqrt{n}$	$\in \Delta(\ln n)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(f)	$(0.5n)^{\log_2 n}$	$\in \Delta\left(2^{(\log_2 n)^2}\right)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.

Begründen Sie Ihre Antworten jeweils mit einem kurzen Beweis.

Allgemeine Hinweise zur Hausaufgabenabgabe

Die Hausaufgabenabgabe erfolgt ausschließlich über Moodle. Bitte vergewissern Sie sich nach der Abgabe selbstständig, dass alle Dateien erfolgreich auf Moodle eingestellt wurden. Programmieraufgaben sind in Java zu lösen. Quelltexte sind als in UTF-8 kodierte `.java`-Textdateien abzugeben; die Dateien dürfen sich in einem Archiv im Zip-Format befinden. Schriftliche Abgaben müssen in Form einer einzigen PDF-Datei erfolgen. Nutzen Sie z.B. ImageMagick[1], um aus eingescannten Bildern ein PDF zu erzeugen. Achten Sie bei eingescannten Bildern darauf, dass diese im PDF-Dokument eine korrekte Orientierung haben. **Abgaben, die sich nicht in beschriebenen Formaten befinden, können nicht korrigiert oder bewertet werden!**

Aufgabe 2.4 [7 Punkte] **(H) Wachs-TUM**

Beweisen oder widerlegen Sie die folgenden Aussagen für **positivwertige** Funktionen f, g, h.

a) $f(n) + g(n) \in \mathcal{O}(h(n))$ impliziert $f(n) \in \mathcal{O}(h(n))$ *und* $g(n) \in \mathcal{O}(h(n))$

b) $f(n) + g(n) \in \mathcal{O}(h(n))$ impliziert $f(n) \in \mathcal{O}(h(n))$ *oder* $g(n) \in \mathcal{O}(h(n))$

c) $f(n) - g(n) \in \mathcal{O}(h(n))$ impliziert $f(n) \in \mathcal{O}(g(n))$ *oder* $g(n) \in \mathcal{O}(f(n))$

d) Ordnen Sie die folgenden Funktionen aufsteigend bzgl. $\mathcal{O}$-Notation. Eine Begründung ist nicht nötig.

$$(n \log_2 n)^2 \qquad 42 \qquad 1.1^{1.1^n} \qquad n^3 3^{3n} \qquad n^{15} \qquad 0.1n \qquad (\log_2 n)^7$$

Zeigen oder widerlegen Sie die folgenden Aussagen:

e) Für jedes $n \in \mathbb{N}$ mit $n > 0$ gilt:

$$\sum_{j=0}^{n} \frac{j}{2^j} \leq 3 - \frac{2n}{2^n}$$

f) Es gilt:

$$\sum_{j=0}^{n} \frac{j}{2^j} \in \mathcal{O}(1)$$

Aufgabe 2.5 [7 Punkte] **(H) Wachstümer**

Beweisen oder widerlegen Sie die folgenden Aussagen. f und g sind dabei positivwertige Funktionen.

a) $\mathcal{O}(f(n)) \cup \mathcal{O}(g(n)) \subseteq \mathcal{O}(f(n) + g(n))$

b) $\mathcal{O}(f(n) + g(n)) \subseteq \mathcal{O}(f(n)) \cup \mathcal{O}(g(n))$

Zeigen oder widerlegen Sie die folgenden Aussagen:

c) $n \in \Omega(n \log_2 n)$

[1] `http://www.imagemagick.org/script/index.php`

d) $(\frac{n}{2})^5 \in o(3n^5)$

e) $n^2 + 8n \in \mathcal{O}(3n^2)$

f) $0.1n + \sqrt[3]{27n^3} \in \Theta(n)$

Aufgabe 2.6 [6 Punkte] **(H) Binäre Suche**

In dieser Aufgabe geht es darum, durch binäre Suche Wertebereiche in Feldern zu finden. Ein Progammgerüst des in Java zu implementierenden Programms wird als separates Archiv mit diesem Übungsblatt verteilt. Kommentare im gegebenen Quelltext enthalten wichtige Hinweise und Beispiele.

a) Implementieren Sie die Methode `search(int[], int, boolean)` der Klasse `BinSea`. Die Methode sucht mittels binärer Suche für eine der beiden Intervallgrenzen einen passenden Index im sortierten Feld. Gibt es einen solchen Index nicht, gibt es die Methode -1 zurück.

b) Implementieren Sie die Methode `search(int[], Interval)` der Klasse `BinSea`. Die Methode sucht für ein gegebenes sortiertes Feld und einen Wertebereich einen Indexbereich, sodass alle Feldwerte des Indexbereiches innerhalb des Wertebereiches liegen. Wird kein passender Indexbereich gefunden, soll das leere Intervall zurückgegeben werden.

Bitte beachten Sie, dass eine Fehlerbehandlung nicht verlangt wird; es ist allerdings oft hilfreich, auf Fehlersituationen sinnvoll zu reagieren, da man auf diese Weise Programmierfehler leichter finden kann.