

Musterlösung

(Mathematik für Informatiker II, SoSe 2007, 7. Aufgabenblatt)

von Sebastian Böthin, Frank Hoffmann

zu Aufgabe 1

Wir benutzen einen einfachen Fakt aus der Physik:

Wenn ein Körper mit Masse M_1 seinen Schwerpunkt in $m_1 \in \mathbb{R}$ hat und ein anderer mit Masse M_2 in m_2 , so ist der gemeinsame Schwerpunkt in m mit

$$m = \frac{M_1 \cdot m_1 + M_2 \cdot m_2}{M_1 + M_2}$$

Stabilität des Stapels: Damit der Stapel bestehend aus n möglichst weit nach rechts überragenden Brettern der Länge 2 (und o.B.d.A. Masse 1) nicht umfällt, ist es notwendig und hinreichend, dass für alle $1 \leq i \leq n - 1$ gilt:

Der gemeinsame Schwerpunkt der obersten $i - 1$ Bretter darf nicht rechts vom rechten Rand des i -ten Brettes von oben liegen, im Grenzfall liegt er also genau darüber.

Wenn wir $i - 1$ Bretter mit ihrem Schwerpunkt genau auf den rechten Rand des i -ten Brettes legen und sei dieser Rand bei 1, so ist der neue gemeinsame Schwerpunkt der i Bretter bei $((i - 1) \cdot 1 + 1 \cdot 0)/i = (i - 1)/i$.

Im nächsten Schritt legen wir diese i Bretter mit ihrem Schwerpunkt auf den rechten Rand des nächsten Brettes und sei dieser Rand wieder bei 1. Damit erhöhen wir den Überhang um $1/i$.

Für den Gesamtüberhang s_n bei n Brettern ergibt sich

$$s_n = \frac{1}{n-1} + \frac{1}{n-2} + \dots + \frac{1}{2} + 1$$

Es handelt sich also um die harmonischen Zahlen, s. Vorlesung. Diese wachsen wie der natürliche Logarithmus, also unbeschränkt.

Mit Maple erhält man $s_{50} = 4,4792$ und $s_{100} = 5,1773$.

Hinweis: Für die Lösung ist entscheidend, dass man immer nur EIN Brett auf das darunterliegende packt. Lässt man diese Forderung fallen, so kann man mit n Brettern nicht nur einen Überhang der Größenordnung $\ln n$ produzieren sondern sogar $\Theta(n^{1/3})$.

Das wurde erst im letzten Jahr bewiesen, siehe

<http://3dpancakes.typepad.com/ernie/2006/01/overhang.html>

zu Aufgabe 2

Wir betrachten auf natürlichen Zahlen definierte positive Funktionen $f(n), g(n)$.
Zunächst erinnern wir uns an die Definitionen:

$$g(n) = O(f(n)) : \iff \exists C > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \leq C f(n) \\ \iff g(n)/f(n) \text{ ist beschränkt.}$$

$$g(n) = o(f(n)) : \iff \forall C > 0 : \exists n_0 : \forall n \geq n_0 : g(n) \leq C f(n) \\ \iff g(n)/f(n) \text{ konvergiert gegen 0.}$$

$$g(n) = \Theta(f(n)) : \iff \exists C_1, C_2 > 0 : \exists n_0 : \forall n \geq n_0 : C_1 f(n) \leq g(n) \leq C_2 f(n) \\ \iff g(n) = O(f(n)) \text{ und } f(n) = O(g(n))$$

Speziell bezeichnet also $O(1)$ die Klasse aller beschränkten Funktionen und $o(1)$ die Klasse aller Nullfolgen.

Hinweis: Man kann sich überlegen, dass die Mengen der Form $\Theta(f(n))$ eine disjunkte Zerlegung der Menge aller betrachteten Funktionen bilden, also Äquivalenzklassen. Auf diesen Klassen definiert die O -Notation eine totale Ordnung \leq in folgendem Sinne:

$$\Theta(g(n)) < \Theta(f(n)) \iff g(n) = o(f(n)), \\ \Theta(g(n)) \leq \Theta(f(n)) \iff g(n) = O(f(n)), \\ \Theta(g(n)) = \Theta(f(n)) \iff g(n) = \Theta(f(n)).$$

Nun sollen folgende Aussagen auf Richtigkeit geprüft werden, d.h. wir haben die jeweilige Aussage entweder zu beweisen oder die Allgemeingültigkeit durch ein Gegenbeispiel zu widerlegen.

(a) $f(n) + g(n) = \Theta(\max(f(n), g(n)))$: Dies ist richtig, denn f und g sind positive Funktionen und damit gilt für jedes n :

$$\max\{f(n), g(n)\} < f(n) + g(n) \leq 2 \cdot \max\{f(n), g(n)\}$$

Wir haben also: $\max\{f(n), g(n)\} = O(f(n) + g(n))$, $f(n) + g(n) = O(\max\{f(n), g(n)\})$

Die Intuition sollte einem sofort sagen, dass dies i.A. falsch ist, wenn man das \max durch ein \min ersetzt. Tatsächlich ist dann bereits $f(n) = n$ und $g(n) = 1$ ein Gegenbeispiel, da

$$n + 1 \neq \Theta(1) = \Theta(\min(n, 1)).$$

(b) $f(n) = \Theta(f(n/2))$:

Für Polynome ist die Aussage offenbar richtig. Probieren wir aber eine andere Klasse von Funktionen aus, etwa $f(n) = 2^n$, so erhalten wir

$$2^{n/2} = (\sqrt{2})^n,$$

und man erinnert man sich nun entweder an die Regel

$$a < b \implies a^n = o(b^n),$$

woraus $(\sqrt{2})^n = o(2^n)$ und daher $2^n \neq O(2^{n/2})$ folgt, oder man betrachtet einfach den Quotienten:

$$2^n / 2^{n/2} = 2^{n-n/2} = 2^{n/2}.$$

Diese Folge ist offenbar nicht beschränkt, daher gilt $2^n \neq O(2^{n/2})$ und erst recht $2^n \neq \Theta(2^{n/2})$.

(c) $f(n) + o(f(n)) = \Theta(f(n))$:

Dies ist eine abkürzende Schreibweise für die Aussage

$$g(n) = o(f(n)) \implies f(n) + g(n) = \Theta(f(n)).$$

Sie bedeutet, dass sich eine Summe von Funktionen in der O -Notation stets auf einen bzgl. O -Notation dominierenden Summanden reduziert. Das entspricht dem gewohnten Umgang mit Summen in O -Notation, und es sollte sich sogar die etwas stärkere Aussage

$$f(n) + O(f(n)) = \Theta(f(n))$$

beweisen lassen:

$g(n) = O(f(n))$ bedeutet, dass $g(n)/f(n)$ beschränkt ist, d.h. es gibt Konstanten $0 \leq C_1 \leq C_2$ so dass

$$C_1 \leq g(n)/f(n) \leq C_2$$

(sogar für alle n). Dann ist

$$1 + C_1 \leq 1 + \frac{g(n)}{f(n)} = \frac{f(n) + g(n)}{f(n)} \leq 1 + C_2,$$

also

$$(1 + C_1)f(n) \leq f(n) + g(n) \leq (1 + C_2)f(n)$$

mit $1 + C_1, 1 + C_2 > 0$, was zu beweisen war. Die Aussage ist also richtig.

(d) $f(n) = O(g(n))$ impliziert $2^{f(n)} = O(2^{g(n)})$:

In der eleganten Schreibweise von (c) könnte man diese Aussage auch in der Form

$$2^{O(g(n))} = O(2^{g(n)})$$

aufschreiben. Das ist zunächst nicht ganz abwegig, da ja $x \leq y$ immer $2^x \leq 2^y$ impliziert. Die Frage ist nun, ob diese Monotonie erhalten bleibt, wenn man von Funktionen $f(n)$ zu Klassen $\Theta(f(n))$ übergeht. Wenn es ein Gegenbeispiel gibt, dann sollte man es unter den Funktionen mit $f(n) > g(n)$ und $f(n) = O(g(n))$ suchen. Der erste Versuch, $f(n) = n+1$ und $g(n) = n$, scheitert. Versuchen wir $f(n) = 2n$ und $g(n) = n$, dann haben wir tatsächlich noch $f(n) = O(g(n))$, aber

$$2^{2n} = 4^n \neq O(2^n)$$

wie in (b). Also ist die Aussage falsch.

Übrigens erhält man jede Menge Gegenbeispiele, wenn man zu irgendeinem $g(n) \neq O(1)$ ein $h(n) \neq O(1)$ mit $h(n) = O(g(n))$ wählt. In (c) haben wir gezeigt, dass für $f(n) = g(n) + h(n)$ dann $f(n) = O(g(n))$ gilt, während

$$2^{f(n)} = 2^{g(n)+h(n)} = 2^{g(n)} 2^{h(n)} \neq O(2^{g(n)}),$$

da $2^{h(n)}$ nicht beschränkt ist.

zu Aufgabe 4

Es sind die Funktionen $a(n), \dots, g(n)$ gemäß O -Notation zu sortieren, und wir führen gleich einige Vereinfachungen durch:

$$\begin{aligned} a(n) &= \sqrt{n}, \\ b(n) &= \log_{10} n = \Theta(\log n), \\ c(n) &= n \log_2 n = \Theta(n \log n), \\ d(n) &= 6n^2 - 36n - 444 = \Theta(n^2), \\ e(n) &= 1000n = \Theta(n), \\ f(n) &= \log_2 n = \Theta(\log n), \\ g(n) &= \lfloor n \log_5 n \rfloor + \lfloor \sqrt{n} \rfloor = \Theta(n \log n). \end{aligned}$$

Wegen der Regel

$$\log_b a = \frac{1}{\log_c b} \log_c a$$

unterscheiden sich Logarithmen zu verschiedenen Basen nur um eine Konstante, d.h. man kann die Angabe der Basis in O -Notation ignorieren. Aus 1(c) folgt außerdem, dass für Polynome immer gilt

$$a_0 + a_1 n + \dots + a_m n^m = \Theta(n^m) \quad (a_m \neq 0).$$

Die Gaussklammern $\lfloor \dots \rfloor$ in $g(n)$ brauchen in O -Notation nicht beachtet zu werden. Wir wissen außerdem, dass

$$\lim_{n \rightarrow \infty} \frac{\log n}{\sqrt{n}} = 0,$$

also $\log n = o(\sqrt{n})$, und natürlich gilt

$$\Theta(\sqrt{n}) < \Theta(n) < \Theta(n \log n) < \Theta(n^2).$$

Also lautet die Antwort:

$$\Theta(b(n)) = \Theta(f(n)) < \Theta(a(n)) < \Theta(e(n)) < \Theta(c(n)) = \Theta(g(n)) < \Theta(d(n))$$