Algorithmen und Datenstrukturen: Übung 1

Alexander Waldenmaier

12. November 2020

Aufgabe 1.1

Zur Lösung der Gleichungen verwende ich einen Taschenrechner.

a)

$$50000000n \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 20$$
$$50000000n \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 1200$$

b)

$$10^7 n \log_2 n \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 22$$
$$10^7 n \log_2 n \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 643$$

c)

$$10^6 n^2 \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 31$$
$$10^6 n^2 \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 244$$

d)

$$1.5^n \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 51$$
$$1.5^n \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 61$$

e)

$$2^n \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 29$$
$$2^n \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 35$$

f)

$$n! \overset{!}{\leq} 1 \cdot 10^9 \Rightarrow n_{max} = 12$$
$$n! \overset{!}{\leq} 60 \cdot 10^9 \Rightarrow n_{max} = 13$$

Aufgabe 1.2

a)

$$\log\left(n^n\right) = n\log n = \mathcal{O}\left(n\log n\right)$$

Die Aussage stimmt.

b)

$$\log_a n = \frac{\log_b n}{\log_b a} = c\cdot\log_b n = \Theta\left(\log_b n\right), c\in\mathbb{R}$$

Die Aussage stimmt.

c) Zu beweisen für beliebige Basen c:

$$\begin{aligned} a^{\log_c b} &= b^{\log_c a} \\ \Leftrightarrow a^{\frac{\log_b b}{\log_b c}} &= b^{\frac{\log_b a}{\log_b c}} \\ \Leftrightarrow a^{\log_b b} &= b^{\log_b a} \\ \Leftrightarrow a &= a, \textbf{q.e.d.} \end{aligned}$$

Demnach kann folgende Umformung durchgeführt werden:

$$2^{\log\left(n^2\right)} = n^{2^{\log 2}}$$

Da nicht bekannt ist, zu welcher Basis der Logarithmus definiert ist, kann keine Aussage gemacht werden.

d) Einsetzen der Bedingung liefert:

$$\begin{aligned} \log(f(n)) = \log(\mathcal{O}(g(n))) &\overset{!}{=} \mathcal{O}(\log(g(n))) \\ \log(c\cdot g(n)) &\overset{!}{=} \mathcal{O}(\log(g(n))), c>0 \\ \log(g(n)) + \log(c) &= \mathcal{O}(\log(g(n))) \end{aligned}$$

Die Aussage stimmt.

e) Zu widerlegen:

$$\exists c>0\ \exists n_0>0\ \forall n\geq n_0 : 2^n \leq c\cdot\sqrt{2}^n$$

Widerspruchbeweis:

$$\begin{aligned} 2^n &\overset{!}{\leq} c\cdot\sqrt{2}^n \\ \left(\frac{2}{\sqrt{2}}\right)^n &\overset{!}{\leq} c \\ n\log\left(\frac{2}{\sqrt{2}}\right) &\overset{!}{\leq} \log c \\ n &\leq \frac{\log c}{\log 2/\sqrt{2}} = n_1 = \text{const.} \end{aligned}$$

Die Aussage wurde widerlegt, da 2^n für alle $n > n_1$ größer $\sqrt{2}^n$ ist.

f) Unter Verwendung der Regel, die in 1.2c) hergeleitet wurde gilt:

$$\begin{aligned}
f(n) &= \log(n)^{\log(n)} \\
&= n^{\log(\log(n))} \overset{!}{\leq} c \cdot n^k \\
\Rightarrow n^{\log(\log(n))-k} &\overset{!}{\leq} c \\
(\log(\log(n)) - k) \log n &\overset{!}{\leq} \log c \\
\log(\log(n)) - k &\overset{!}{\leq} \frac{\log c}{\log n} \\
\log(\log(n)) - \frac{\log c}{\log n} &\overset{!}{\leq} k \\
\lim_{n\to\infty} \log(\log(n)) - \frac{\log c}{\log n} &\overset{!}{\leq} k \\
\infty &\overset{!}{\leq} k
\end{aligned}$$

Die Aussage wurde widerlegt. Es existiert kein k für das gilt: $f(n) = \mathcal{O}\left(n^k\right)$.

Aufgabe 1.3

a)

$$\begin{aligned}
T_1(n) &= 5T_1\left(\frac{n}{3}\right) + T_1\left(\frac{2n}{3}\right) + 3n, m = 2, k = 1 \\
&\sum_{i=1}^{m} \alpha_i^k = \frac{5}{3} + \frac{2}{3} = \frac{7}{3} > 1 \Rightarrow \text{Fall 3!} \\
&\sum_{i=1}^{m} \alpha_i^c = 1 \\
&\Leftrightarrow 5\left(\frac{1}{3}\right)^c + \left(\frac{2}{3}\right)^c = 1 \\
&\Rightarrow c = 2 \\
&\Rightarrow T_1(n) = \Theta(n^2)
\end{aligned}$$

b)

$$\begin{aligned}
T_1(n) &= T_2\left(\frac{n}{4}\right) + 2T_2\left(\frac{n}{16}\right) + \sqrt{n} \\
&= T_2\left(\frac{n}{4}\right) + 2T_2\left(\frac{n}{16}\right) + n^{1/2}, m = 2, k = \frac{1}{2} \\
&\sum_{i=1}^{m} \alpha_i^k = \sqrt{\frac{1}{4}} + 2\sqrt{\frac{1}{16}} = \frac{1}{2} + \frac{1}{2} = 1 \Rightarrow \text{Fall 2!} \\
&\Rightarrow T_2(n) = \Theta(\sqrt{n} \log n)
\end{aligned}$$

c)

$$\begin{aligned}
T_3(n) &= T_3\left(\frac{3n}{4}\right) + 2T_3\left(\frac{n}{16}\right) + 4n, m = 2, k = 1 \\
\sum_{i=1}^{m} \alpha_i^k &= \frac{3}{4} + 2\frac{1}{16} = \frac{7}{8} < 1 \Rightarrow \text{Fall 1!} \\
&\Rightarrow T_3(n) = \Theta(n)
\end{aligned}$$

Aufgabe 1.4

Für das kleinste α gilt:

$$\begin{aligned}
f(n) = 2 \cdot f(n-1) + f(n-2) &\overset{!}{=} \alpha^n \\
\Rightarrow 2 \cdot \alpha^{n-1} + \alpha^{n-2} &= \alpha^n \\
2\alpha + 1 &= \alpha^2 \\
\alpha^2 - 2\alpha - 1 &= 0 \\
\Rightarrow \alpha_1 = 1 - \sqrt{2} < 0, \alpha_2 = 1 + \sqrt{2} &> 0
\end{aligned}$$

Mit den Bedingungen $f(1) = 1, f(2) = 2$ ergibt sich, dass Lösung α_2 die richtige ist, um $f(n) \leq \alpha^n$ zu erfüllen.

Aufgabe 1.5

a) Betrachten wir das Spiel mit den Stäben A (Start), B (Zwischenspeicher), C (Ziel) und einer Anzahl von Scheiben n.

Im Fall $n = 1$ ist die Aufgabe einfach: Die Scheibe wird von A nach C bewegt und das Spiel ist in einem Zug vorbei.

Im Fall $n = 2$ wird erstmals der Zwischenspeicher verwendet: Man bewegt zunächst die kleinste Scheibe von A nach B. Jetzt befindet man sich wieder in der Situation vom Anfang, in der eine einzelne Scheibe (die größere) von A nach C bewegt werden muss. Abschließend wird noch die kleine Scheibe von B nach C bewegt, und das Spiel ist nach 3 Zügen vorbei.

Im Fall $n = 3$ erkennt man, wie ein rekusives Muster ensteht: Die erste Aufgabe besteht darin, die Pyramide der obersten zwei Scheiben von A nach B zu bewegen. Dies geschieht in 3 Zügen, wie im Fall $n = 2$. Anschließend kann die größte Scheibe einfach von A nach C bewegt werden und befindet sich damit an ihrem finalen Platz (+1 Zug). Nun besteht die Aufgabe erneut darin, die Pyramide mit zwei Scheiben von B nach C, unter Hilfe von A als Zwischenspeicher, zu bewegen. Dies geschieht erneut in 3 Zügen. Damit sind insgesamt 7 Zügen notwendig gewesen.

Generell lässt sich das Problem also wie folgt formulieren: Die benötigte Anzahl an Zügen $B(n)$ für n Scheiben entspricht der zweifachen Anzahl an Zügen bei $n-1$ Scheiben, plus 1, um die Scheibe selbst zu bewegen:

$$\begin{aligned} B(1) &= 1 \\ B(n) &= 2 \cdot B(n-1) + 1 \end{aligned}$$

Als Ansatz für die explizite Lösung wählen wir:

$$\begin{aligned} \text{I: } B(n) = 2 \cdot B(n-1) + 1 &\stackrel{!}{=} a^n + c \\ 2 \cdot B(n-1) + 1 &\stackrel{!}{=} a^n + c \\ \text{mit I } \Rightarrow 2\left(a^{n-1} + c\right) + 1 &\stackrel{!}{=} a^n + c \\ 2a^{-1}a^n + 2c + 1 &\stackrel{!}{=} a^n + c \end{aligned}$$

Koeffizientenvergleich liefert:

$$\begin{aligned} 2a^{-1} = 1 &\Rightarrow a = 2 \\ 2c + 1 = c &\Rightarrow c = 1 \\ &\Rightarrow B(n) = 2^n - 1 \end{aligned}$$

b) Abgabe in DOMjudge. Teamname: "test"
Die Aufgabe war verdammt schwer, zumindest in Anbetracht der Punktzahl! Ging das nur mir so?