Algorithms & Data Structures Solutions - SoSe 24

Igor Dimitrov

2024-04-22

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Blatt 01

1.1 Aufgabe 2

a)

$$\begin{aligned}
\log(n!) &= \log(\prod_{i=1}^{n} i) && \text{(Def } n!\text{)} \\
&= \sum_{i=1}^{n} \log(i) && \text{(Eig } \log(\bullet)\text{)} \\
&\leq \sum_{i=1}^{n} \log(n) && \text{(Eig } \log(\bullet)\text{)} \\
&= n \log(n)
\end{aligned}$$

Waehle nun $c_0 := 1$ und $n_0 := 1$. Es folgt somit:

$$\begin{aligned}
& \log(n!) \leq 1 \cdot n \log(n), \quad \forall n \geq 1 \\
\iff & \log(n!) \in \mathcal{O}(n \log(n)) && \text{(Def } \mathcal{O}\text{)}
\end{aligned}$$

b)

Zuerst bemerken wir die folgende Eigenschaft

$$\begin{aligned}
n \log(n) &\leq c \log(n!) \\
\iff c &\geq \frac{n \log(n)}{\log(n!)} \\
&= \frac{n \log(n)}{\sum_{i=1}^{n} \log(i)}
\end{aligned}$$

Wir definieren die Folge:

$$
\begin{aligned}
c(n) &:= \frac{n\log(n)}{\sum_{i=1}^{n}\log(i)} \\
&= \frac{\overbrace{\log(n)+\dots+\log(n)}^{\text{n-mal}}}{\log(1)+\dots+\log(n)}
\end{aligned}
$$

Wir behaupten ohne Beweis, dass $c(n)$ eine monoton fallende Folge ist. D.h. es gilt:

$$c(n) \leq c(m), \quad \forall n \geq m$$

Setze nun $n_0 := 10, c_0 := c(10) = \frac{10\log(10)}{\sum_{i=1}^{10}\log(i)}$. Somit folgt:

$$
\begin{aligned}
n\log(n) &\leq \left(\frac{n\log(n)}{\sum_{i=1}^{n}\log(i)}\right)\log(n!) \\
&= c(n)\log(n!) \\
&\leq c_0\log(n!), \quad \forall n \geq n_0 = 10 && (c(n) \text{ monoton fallend})
\end{aligned}
$$

1.2 Aufgabe 3

a)

Da $f_1 \in \mathcal{O}(g_1)$ und $f_2 \in \mathcal{O}(g_2)$ existieren n_1, n_2, c_1, c_2 s.d:

$$
\begin{aligned}
f_1(n) &\leq c_1 g_1(n), \quad \forall n \geq n_1 \\
f_2(n) &\leq c_2 g_2(n), \quad \forall n \geq n_2
\end{aligned}
$$

Setze $c_0 := max\{c_1, c_2\}, n_0 := max\{n_1, n_2\}$. Dann gilt

$$
\begin{aligned}
(f_1+f_2)(n) &= f_1(n) + f_2(n) \\
&\leq c_1 g_1(n) + c_2 g_2(n), \quad \forall n \geq n_0 && (n \geq n_0 \Rightarrow n \geq n_1 \land n \geq n_2) \\
&\leq c_0 g_1(n) + c_0 g_2(n), \quad \forall n \geq n_0 \\
&= c_0(g_1+g_2)(n) \quad \forall n \geq n_0 \\
\iff \; f_1+f_2 &\in \mathcal{O}(g_1+g_2) && (\text{Def } \mathcal{O})
\end{aligned}
$$

b)

mit $f_1 \in \Theta(g_1), f_2 \in \Theta(g_2)$ existieren $a_1, a_2, b_1, b_2, n_1, n_2$, s.d.:

$$a_1 f_1(n) \leq g_1(n) \leq a_2 f_1(n), \forall n \geq n_1$$
$$b_1 f_2(n) \leq g_2(n) \leq b_2 f_2(n), \forall n \geq n_2$$

Setze $n_0 := max\{n_1, n_2\}$, $c_1 := a_1 b_1$, $c_2 := a_2 b_2$. Dann gilt:

$$c_1(f_1 f_2)(n) = a_1 f_1(n) a_2 f_2(n) \leq (g_1 g_2)(n) \leq b_1 f_1(n) b_2 f_2(n) = c_2(f_1 f_2)(n), \quad \forall n \geq n_0$$

Somit $f_1 f_2 \in \Theta(g_1 g_2)$.

c) **Falsch**:

Betrachte $f(n) := n$ und $g(n) := 10n$. Offensichtlicht gilt $f \in \Omega(g)$ mit $c_0 := 1/10, n_0 := 1$. Aber $2^n \notin \Omega(2^{10n})$, da 2^n langsamer als 2^{10n} waechst. (Setze z.B. $2^n := x$. Dann $2^{10n} = (2^n)^{10} = x^{10}$, und x^{10} ist offensichtlich schneller als x)

d) **Falsch**:

Sei $g(n) := 2^n$. Dann $f(n) = g(2n) = 2^{2n} = (2^n)^2$. $(2^n)^2$ ist offensichtlich schneller als 2^n

e) **Falsch**:

Seien $f(n) := n^2, f_1(n) := n^3, f_2(n) := n$. Es gilt:

$$f \in \mathcal{O}(f_1) \qquad (n^2 \in \mathcal{O}(n^3))$$
$$f_1 \in \Omega(f_2) \qquad (n^3 \in \Omega(n))$$

aber

$$f \notin \mathcal{O}(f_2) \qquad (n^2 \notin \mathcal{O}(n))$$

f)

Es gilt:

$$
\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{f_2}(n) &= \lim_{n\to\infty} \left(\frac{f(n)}{f_1(n)} \cdot \frac{f_1(n)}{f_2(n)} \right) \\
&= \lim_{n\to\infty} \left(\frac{f(n)}{f_1(n)} \right) \cdot \lim_{n\to\infty} \left(\frac{f_1(n)}{f_2(n)} \right) \\
&= 0 \cdot c, \text{fuer ein} c && (f \in o(f_1), f_1 \in \mathcal{O}(f_2)) \\
&= 0 \\
\iff f \in o(f_2) & && (\text{Def } o)
\end{aligned}
$$

Wobei wir die alternativen Definitionen von $o(\bullet)$ und $\mathcal{O}(\bullet)$ benutzt haben.