Gruppe $\boxed{\textbf{14}}$
Datenstrukturen und Algorithmen

Phil Pützstück, 377247
Benedikt Gerlach, 376944
Sebastian Hackenberg, 377550

Hausaufgabe 2

Aufgabe 1

a)

Es sei f eine gegebene Funktion. Da $\forall n \in \mathbb{N}_0 : 1 \cdot f(n) = f(n)$ existiert eine konstante $0 < c = 1 < \infty$ und ein $n_0 = 0$ sodass gilt:

$$\forall n \in \mathbb{N}_0, n \geq n_0 : f(n) \leq c \cdot f(n)$$

Damit haben wir nach Definition $f \in \mathcal{O}(f) \iff f \sqsubseteq f$. Also ist $\sqsubseteq$ reflexiv.

Ferner seien nun weiter Funktionen g, h gegeben. Es gilt nach Definition:

$$(f \sqsubseteq g) \wedge (g \sqsubseteq h) \implies (f \in \mathcal{O}(g)) \wedge (g \in \mathcal{O}(h))$$

$$\implies (\exists c, n_0 : \forall n \geq n_0 : f(n) \leq c \cdot g(n)) \wedge (\exists c', n_0' : \forall n \geq n_0 : g(n) \leq c \cdot h(n))$$

Sei nun $n' := \max\{n_0, n_0'\}$. Dann folgt weiter:

$$\exists c, c' : \forall n \geq n' : f(n) \leq c \cdot g(n) \quad \wedge \quad g(n) \leq c' \cdot h(n)$$

$$\implies \exists c, c' : \forall n \geq n' : f(n) \leq c \cdot g(n) \leq c(c' \cdot h(n))$$
$$\implies \exists c, c' : \forall n \geq n' : f(n) \leq (c \cdot c') \cdot h(n))$$
$$\implies f \in \mathcal{O}(h) \iff f \sqsubseteq h$$

Wir haben also gezeigt, dass für beliebige Funktionen f, g, h gilt:

$$f \sqsubseteq g \wedge g \sqsubseteq h \implies f \sqsubseteq h$$

Damit ist $\sqsubseteq$ reflexiv und transitiv, also eine Quasiordnung $\square$

b)

Es gilt:

$$0 \sqsubseteq 4 \sqsubseteq 2^{9000} \sqsubseteq \log(n) \sqsubseteq n \cdot \log(n) \sqsubseteq n \cdot \sqrt{n} \sqsubseteq n^2 \sqsubseteq \sum_{i=0}^{n} \frac{14i^2}{1+i} \sqsubseteq n^2 \cdot \log(n) \sqsubseteq \frac{n^3}{2} \sqsubseteq n^3 \sqsubseteq 2^n \sqsubseteq n! \sqsubseteq n^n$$

Aufgabe 2

a)

Nach Definition der Klasse $\mathcal{O}$ gilt g $\in \mathcal{O}(f)$ falls c $\cdot$ f(n) ab einer bestimmten Konstanten $n_0 \in \mathbb{N}$ eine obere Schranke von g(n) ist. Wir beweisen durch Induktion, dass die Aussage:

$$\frac{1}{4}n^3 - 7n + 17 < 1 \cdot (n^3)$$

für $n_0 \geq 2$ gilt.

Induktionsanfang

$$A(n) := \frac{1}{4}n^3 - 7n + 17 < n^3$$

$$A(2) = \frac{1}{4}2^3 - 7 \cdot 2 + 17 = 5 < 8 = 2^3$$

Damit gilt die Annahme für n = 2.

Induktionsvoraussetzung

Es gelte $A(n)$ für ein beliebig aber festes $n \in \mathbb{N}$.

Induktionsschritt $n \to n+1$

$$\begin{aligned} \frac{1}{4}(n+1)^3 - 7(n+1) + 17 &= \frac{1}{4}(n^3 + 3n^2 + 3n + 1) - 7n + 7 + 17 \\ = \frac{1}{4}n^3 - 7n + 17 + \frac{1}{4}(3n^2 + 3n + 1) + 7 &\overset{\text{(IV)}}{<} n^3 + \frac{1}{4}(3n^2 + 3n + 1) + 7 \\ < n^3 + \frac{1}{4}(3n^2 + 3n + 1) + 7 &\overset{\text{Lemma}}{<} n^3 + 3n^2 + 1 = (n+1)^3 \end{aligned}$$

$\square$

Lemma

Um die Ungleichung zu Beweisen benutzen wir vollständige Induktion.

Umformung

$$\frac{1}{4}(3n^2 + 1) + 7 < 3n^2 + 3n + 1$$

$$\Leftrightarrow 3n^2 + 3n < 12n^2 + 12n - 24$$

Induktionsanfang

$$A(n) := 3n^2 + 3n < 12n^2 + 12n - 24, n \in \mathbb{N}, n \geq 2$$

$$A(2) = 3 \cdot 2^2 + 3 \cdot 2 = 30 < 48 = 12 \cdot 2^2 + 12 \cdot 2 - 24$$

Induktionsvoraussetzung

Es gelte $A(n)$ für ein beliebig aber festes $n \in \mathbb{N}$.

Induktionsschritt $n \to n+1$

$$A(n+1) = 3(n+1)^2 + 3 \cdot (n+1) = 3(n^2 + 2n + 1) + 3n + 3$$

$$= 3n^2 + 3n + 6n + 6 < 12n^2 + 12n - 24 + 6n + 6 \overset{\text{n} \geq 0}{<} 12n^2 + 12n - 24 + 24n + 12n$$

$$= 12n^2 + 24n + 12 + 12n + 12 - 24 = 12 \cdot (n^2 + 2n + 1) + 12n + 12 - 24$$

$$= 12 \cdot (n+1)^2 + 12 \cdot (n+1) - 24$$

□

b)

Da $3n^2 + 4 > 0$ und $n^4 < 2n^4$ für alle $n \geq 0$ gilt, folgt $n^4 < 1 \cdot (2n^4 + 3n^2 + 42)$ für $n \in \mathbb{N}$. Damit existiert ein $c > 0$ sodass $g(n) < c \cdot f(n)$. Also ist $n^4 \in \mathcal{O}(2n^4 + 3n^2 + 42)$.

□

c)

Die Aussage gilt nach der Alternativen Definition, da:

$$\lim_{n\to\infty} \frac{\log 2(n)}{n} \overset{\text{L'Hospital}}{=} \lim_{n\to\infty} \frac{(\log 2(n))'}{(n)'} = \lim_{n\to\infty} \frac{\frac{1}{n \cdot \ln(2)}}{1} = 0$$

□

d)

Nach dem Lemma der Klasse Groß-O muss es ein $c \geq 0$ mit $c \neq \infty$ geben sodass

$$\lim_{n\to\infty} \frac{\log n}{n^\varepsilon}, \forall \varepsilon > 0$$

Es gilt $\lim_{n\to\infty} \log 2(n) = \infty$ und $\forall \varepsilon > 0 : \lim_{n\to\infty} n^\varepsilon = \infty$.

Somit ist nach L'Hospital

$$\lim_{n\to\infty} \frac{(\log 2(n))'}{(n^\varepsilon)'} = 0$$

Damit ist die Aussage für alle $\varepsilon > 0$ wahr.

□

e)

Nach dem Lemma der Klasse Groß-Theta muss es ein $0 < c < \infty$ geben sodass

$$\lim_{n\to\infty} \frac{a^n}{b^n} = c$$

für die zwei beliebigen Konstanten $a, b > 1$ die Aufgabenstellung gilt.
Beweis durch Gegenbeispiel:

$$\lim_{n\to\infty} \frac{2^n}{4^n} = \lim_{n\to\infty} \frac{1}{2n} = 0$$

Somit ist die Aussage falsch, da $c = 0$.

$\square$

Aufgabe 3

a)

Es sei eine Funktion $g(n)$ gegeben. Wir notieren im Sinne der Lesbarkeit

$$o_1 := o(g(n)) \qquad \mathcal{O}_1 := \mathcal{O}(g(n)) \qquad \Theta_1 := \Theta(g(n))$$

Behauptung: Es gilt $o_1 = \mathcal{O}_1 \setminus \Theta_1$ (1)
Beweis:

$$\begin{aligned} o_1 &:= \{f \mid \forall c > 0 : \exists n_0 : \forall n \geq n_0 : 0 \leq f(n) < c \cdot g(n)\} \\ &= \{f \mid \forall c > 0 : \exists n_0 : \forall n \geq n_0 : 0 \leq f(n) < cg(n) \\ &\qquad \wedge \quad \neg(\exists c_1, c_2 > 0, n_0 : \forall n \geq n_0 : c_1 f(n) \leq g(n) \leq c_2 f(n))\} \\ &= \mathcal{O}_1 \setminus \Theta_1 \end{aligned}$$

$\square$

Nun folgt direkt:

$$o_1 \cap \Theta_1 \overset{(1)}{=} (\mathcal{O}_1 \setminus \Theta_1) \cap \Theta_1 = \varnothing$$

$\square$

b) Wir geben ein Gegenbeispiel. Es seien

$$g(n) = 0 \qquad f(n) = n \qquad h(n) = n^2$$

Damit gilt $f(n) \in \Omega(g(n))$, da für $c = 1 > 0$ und $n_0 = 1$ stets $c \cdot g(n) = 0 \leq f(n)$ gilt.
Weiter ist auch $f(n) \in \mathcal{O}(h(n))$, da für $c = 1$ und $n_0 = 1$ stets $f(n) \leq c \cdot h(n)$ gilt.

Wir nehmen nun an, es gelte $g(n) \in \Theta(h(n))$. Also auch $g(n) \in \Omega(h(n))$. Daraus folgt, dass eine Konstante $c > 0$ existiert, sodass ab einem beliebigem Punkt stets $c \cdot h(n) = cn^2 \leq 0 = g(n)$ gilt. Dies kann offensichtlich nur für $c = 0$ gelten, jedoch muss $c > 0$ erfüllt sein. Damit haben wir einen Widerspruch. Also kann die Aussage nicht stimmen. $\square$

Aufgabe 4

a)

Die Best-Case Laufzeit des Algorithmus beträgt B(n)=1 für den Fall, dass die Länge das Arrays $n = 0$ beträgt. Im Fall $n > 0$ beträgt die Best-Case Laufzeit des Algorithmus B(n)=2n+2$\in \mathcal{O}(n)$.

b)

Die Worst-Case Laufzeit des Algorithmus beträgt W(n)=3n+1$\in \mathcal{O}(n)$.

c)

Die Average-Case Laufzeit des Algorithmus beträgt:

$$\sum_{i=0}^{n} \frac{1}{n+1} \cdot (2 \cdot (n-i) + 3i + 1)$$

d)

```
int allTrue(bool [] E) {
    if (E.length < 1) {
        return -1;
    }
    int m = E.length;
    int i = 0;
    while (i < E.length) {
        if (E[i] == false) {
            return 0;
        }
        i = i + 1;
    }
    return 1;
}
```

Der Algorithmus besitzt eine bessere Average-Case Laufzeit, da er bei einem False in dem Array sofort abbricht und nicht noch das ganze restliche Array durchläuft.

e)

Es gibt keinen äquivalenten Algorithmus dessen Worst-Case Laufzeit in o(n) liegt, da er das gesamte Array durchgehen muss um zu bestätigen das kein False vorkommt.