Tutorium Grundlagen: Algorithmen und Datenstrukturen

Übungsblatt Woche 2

Aufgabe 2.1
Induktion 2

Aufgabe 2.1 (a)

Beweisen Sie: $\mathbf{19 | (5 \cdot 2^{3n+1} + 3^{3n+2})}$ **gilt für alle** $\boldsymbol{n} \in \mathbb{N} \coloneqq \{\mathbf{1, 2, 3, 4, 5, \dots}\}$

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

IB: $n=0$: $(5 \cdot 2^{3\cdot 0+1} + 3^{3\cdot 0+2}) = 5 \cdot 2 + 3^2 = 19$ ✓

IV: Für ein beliebiges aber festes $n \in \mathbb{N}_0$ gilt: $19 | (5 \cdot 2^{3n+1} + 3^{3n+2})$

IS: $5 \cdot 2^{3(n+1)+1} + 3^{3(n+1)+2} = 5 \cdot 2^{3n+3+1} + 3^{3n+3+2}$

$$= 5 \cdot 8 \cdot 2^{3n+1} + 27 \cdot 3^{3n+2}$$

$$= 5 \cdot 8 \cdot 2^{3n+1} + 27 \cdot 3^{3n+2} + 27 \cdot (5 \cdot 2^{3n+1}) - 27 (5 \cdot 2^{3n+1})$$

$$= 27 (\underbrace{3^{3n+2} + 5 \cdot 2^{3n+1}}) + 8 \cdot \underbrace{5 \cdot 2^{3n+1}} - 27 \, \underbrace{5 \cdot 2^{3n+1}}$$

$$= 27 (19 \cdot x) - 19 (5 \cdot 2^{3n+1})$$

$$= \underbrace{19} (\underbrace{27x} - \underbrace{5 \cdot 2^{3n+1}})$$

Aufgabe 2.1 (b)

Beweisen Sie: $\left(\sum_{i=0}^{n} i\right)^2 = \sum_{i=0}^{n} i^3$ **gilt für alle** $n \in \mathbb{N} := \{1, 2, 3, 4, 5, \dots\}$

(Hinweis: $1 + 2 + 3 + \cdots + n = \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ könnte nützlich sein)

IB: $n=1$: $\left(\sum_{i=0}^{1} i\right)^2 = 1^2 = 1^3 = \sum_{i=0}^{1} i^3$ ✓

IV: Für ein beliebiges aber festes $n \in \mathbb{N}$ gilt : $\left(\sum_{i=0}^{n} i\right)^2 = \sum_{i=0}^{n} i^3$

IS: $\left(\sum_{i=0}^{n+1} i\right)^2 = \sum_{i=0}^{n+1} i \cdot \sum_{i=0}^{n+1} i$

$$= \left\{(n+1) + \sum_{i=0}^{n} i\right\}\left\{(n+1) + \sum_{i=0}^{n} i\right\}$$

$$= (n+1)^2 + 2(n+1)\sum_{i=0}^{n} i + \left(\sum_{i=0}^{n} i\right)^2$$

$$= (n+1)^2 + \cancel{2}(n+1)\frac{n(n+1)}{\cancel{2}} + \left(\sum_{i=0}^{n} i\right)^2$$

$$= \underbrace{(n+1)^2} + n\underbrace{(n+1)}^2 + \left(\sum_{i=0}^{n} i\right)^2$$

$$\rightarrow (n+1)^2(n+1) + \left(\sum_{i=0}^{n} i\right)^2$$

$$= (n+1)^3 + \sum_{i=0}^{n} i^3$$

$$= \sum_{i=0}^{n+1} i^3$$

Aufgabe 2.2
Laufzeit-Analyse

Aufgabe 2.2

(a) Erklären Sie in natürlicher Sprache, was die Funktionen jeweils berechnen.

(b) Welche asymptotischen Laufzeiten haben die Algorithmen? Begründen Sie ihre Antwort.

Funktion 1

```
int f(int[] A, int[] B) {
  int result = 0;
  for (int a : A) {
    for (int b : B) {
      if (a == b) {
        result += a;
        break;
      }
    }
  }
  return result;
}
```

(a) Summe aller Werte in A für die es ein gleichwertiges Element in B gibt.

(b) Im Worst-Case: Loope für jedes Element in A einmal über ganz B
$\Rightarrow O(|A| \cdot |B|)$ (wobei $|\ldots|$ die Länge des jeweiligen Arrays ist)

Aufgabe 2.2

(a) Erklären Sie in natürlicher Sprache, was die Funktionen jeweils berechnen.

(b) Welche asymptotischen Laufzeiten haben die Algorithmen? Begründen Sie ihre Antwort.

Funktion 2

```
int f(int n) {
  if (n <= 0) return 0;
  if (n == 1) return 1;
  return f(n-1) + f(n-2);
}
```

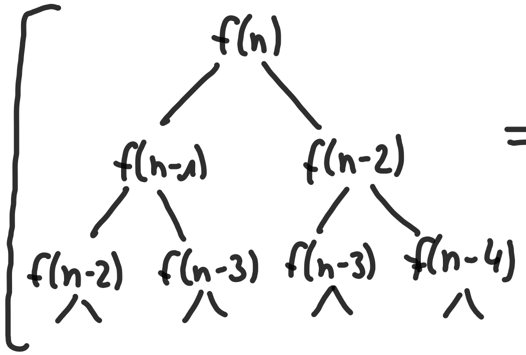
(a) n-te Fibonacci-Zahl

(b) Beachte: konstanter Aufwand pro Funktionsaufruf

$\Rightarrow$ Laufzeit abhängig von der Anzahl rekursiver Aufrufe

maximal n+1 Layers

(Eingabe jedes Mal um min. 1 reduziert)



$\Rightarrow$ Anzahl abschätzen durch Knoten in vollständigem Binärbaum mit Höhe n

$$\Rightarrow O\left(\sum_{i=0}^{n} 2^i\right) = O(2^{n+1}-1) = O(2^{n+1}) = O(2^n \cdot 2) = \underline{\underline{O(2^n)}}$$

Aufgabe 2.2

(a) Erklären Sie in natürlicher Sprache, was die Funktionen jeweils berechnen.

(b) Welche asymptotischen Laufzeiten haben die Algorithmen? Begründen Sie ihre Antwort.

Funktion 3

```
int f(int n) {
  if (n < 0) return -1;
  if (n == 0) return 1;
  return n * f(n-1);
}
```

n+1 Calls

f(n)
|
f(n-1)
|
...
|
f(1)
|
f(0)

(a) Berechnet $n!$ (Fakultät)

(b) Ähnliche Begründung zu Funktion 2

- Konstanter Aufwand pro Funktionsaufruf

$\Rightarrow$ Laufzeit abhängig von Call-Anzahl

- Hier genau bekannte Aufrufzahl: $n+1$ (+1 stammt vom Call mit $n=0$)

$\Rightarrow \mathcal{O}(n+1) = \mathcal{O}(n)$

Aufgabe 2.3
$\mathcal{O}$-Notation

Wiederholung: $\mathcal{O}$-Notation

Mengen des asymptotischen Verhaltens einer Funktion $f : \mathbb{N} \to \mathbb{R}^+$:

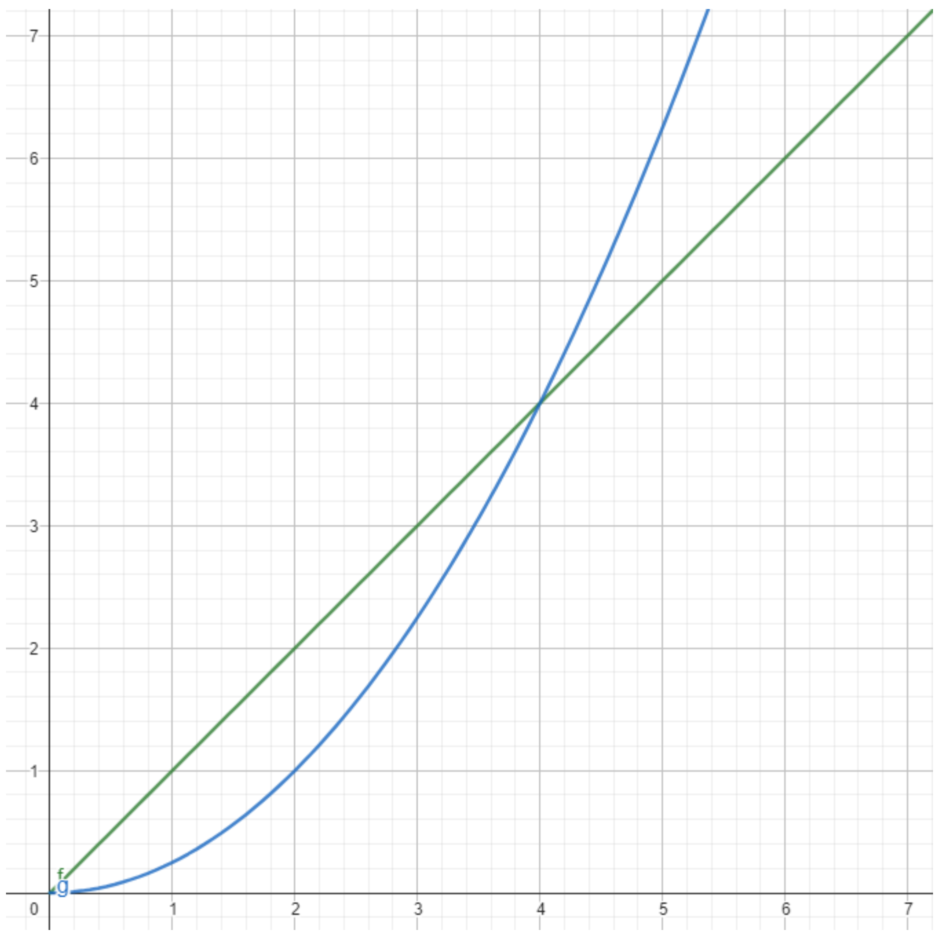
- $\mathcal{O}\big(f(n)\big) = \big\{g(n) \mid \exists c > 0\ \exists n_0 > 0\ \forall n \geq n_0 :\ g(n) \leq c \cdot f(n)\big\}$
  (Funktionen die asymptotisch nicht schneller als f wachsen)
- $\Omega\big(f(n)\big) = \big\{g(n) \mid \exists c > 0\ \exists n_0 > 0\ \forall n \geq n_0 :\ g(n) \geq c \cdot f(n)\big\}$
  (Funktionen die asymptotisch nicht langsamer als f wachsen)

Ein Faktor c reicht aus!!!

- $\Theta\big(f(n)\big) = \mathcal{O}(f(n))\ \cap\ \Omega(f(n))$
  (Funktionen die asymptotisch gleiches Wachstum wie f haben)
- $o\big(f(n)\big) = \big\{g(n) \mid \underbrace{\forall C > 0}\ \exists n_0 > 0\ \forall n \geq n_0 :\ g(n) \leq C \cdot f(n)\big\}$
  (Funktionen die asymptotisch langsamer als f wachsen)
- $\omega\big(f(n)\big) = \big\{g(n) \mid \forall C > 0\ \exists n_0 > 0\ \forall n \geq n_0 :\ g(n) \geq C \cdot f(n)\big\}$
  (Funktionen die asymptotisch schneller als f wachsen)

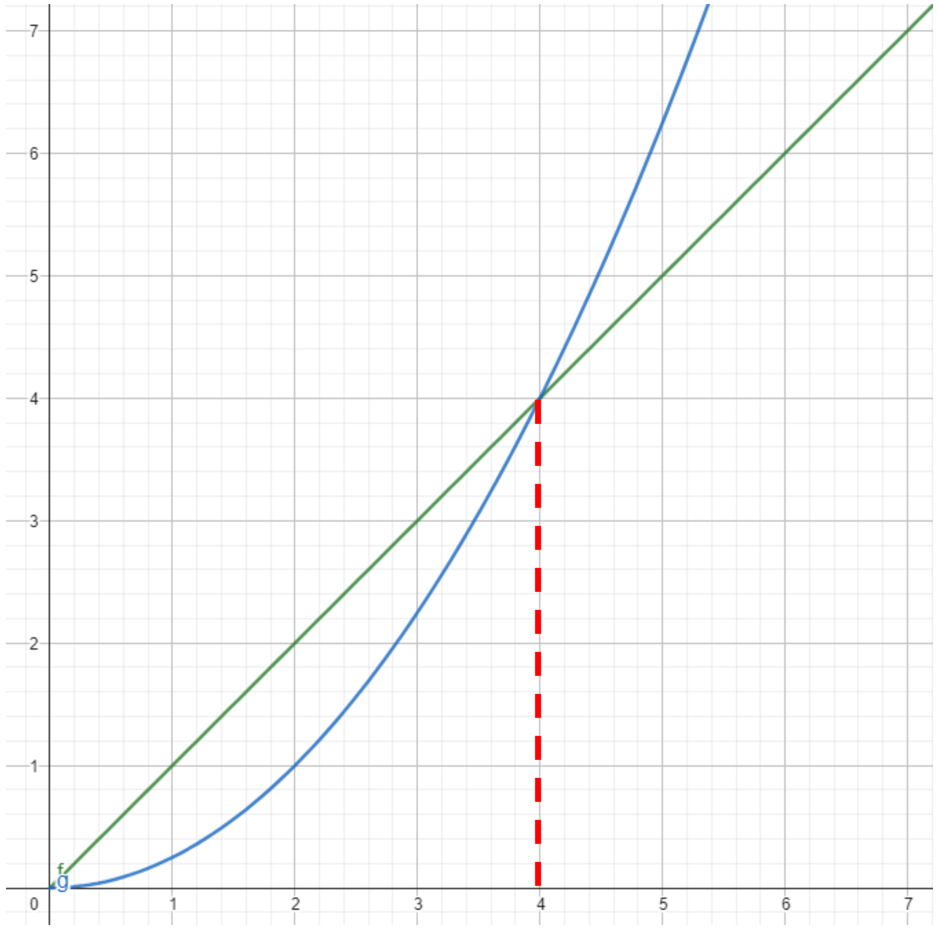
Muss hier für **jedes** mögliche C gelten!!!

Beispiele: $\mathcal{O}$-Notation



Funktion g (blau) wächst offensichtlich schneller

Beispiele: $\mathcal{O}$-Notation



Funktionen:

$$f(x) = x$$
$$g(x) = 0.25x^2$$

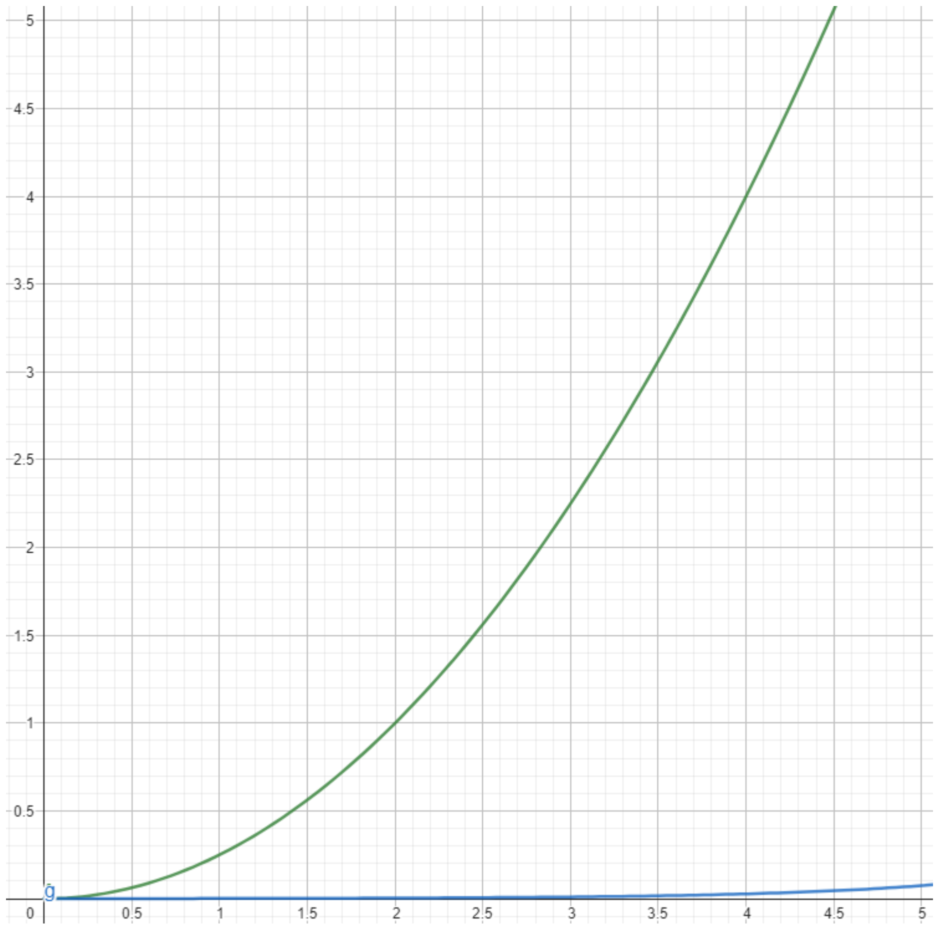
Schnittpunkt:

$$x = 0.25x^2$$
$$1 = 0.25x$$
$$4 = x$$

Limes:

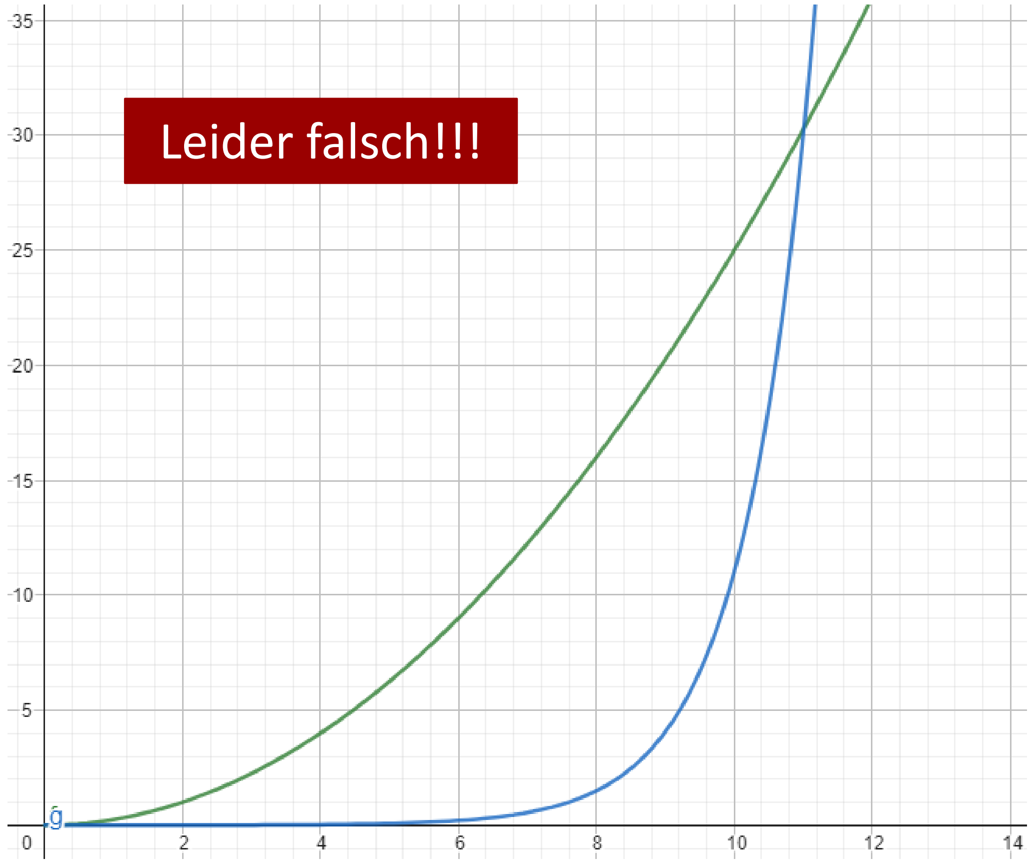
$$\lim_{x\to\infty}\left(\frac{f(x)}{g(x)}\right) = \lim_{x\to\infty}\left(\frac{x}{0.25x^2}\right) = \frac{1}{0.25}\lim_{x\to\infty}\left(\frac{x}{x\text{^}2}\right)$$
$$= 4\lim_{x\to\infty}\left(\frac{1}{x}\right) = 4\cdot 0 = 0 \Rightarrow f(x) \in \mathcal{o}\big(\mathrm{g(x)}\big)$$

Beispiele: $\mathcal{O}$-Notation



Funktion g (blau) wächst offensichtlich langsamer

Beispiele: $\mathcal{O}$-Notation



Funktionen:

$$f(x) = 0.25x^2$$
$$g(x) = 0.0005e^x$$

Schnittpunkt und Ableitung: Eher aufwändig

Grenzwert:

$$\lim_{x\to\infty}\left(\frac{f(x)}{g(x)}\right) = \lim_{x\to\infty}\left(\frac{0.25x^2}{0.0005e^x}\right)$$

$$= \frac{0.25}{0.0005}\lim_{x\to\infty}\left(\frac{x^2}{e^x}\right) = 500\lim_{x\to\infty}\left(\frac{x^2}{e^x}\right)$$

$$\overset{*}{=} 500\lim_{x\to\infty}\left(\frac{2x}{e^x}\right) = 500\lim_{x\to\infty}\left(\frac{2}{e^x}\right) = 500\cdot 0 = 0$$

$$\Rightarrow f(x) \in o(\mathrm{g(x)})$$

* Regel von de L'Hospital

https://de.wikipedia.org/wiki/Landau-Symbole#Definition

Limes Definitionen für O-Notation

Aufgabe 2.3

Kreuzen Sie das am besten passende und stärkste Symbol an welches an Stelle von Δ eingesetzt werden soll.
(Sind mehrere möglich nur das stärkere, z.B. bei o und $\mathcal{O}$ nur o; bei $\mathcal{O}$, Ω und Θ nur Θ)
Begründen Sie ihre Antwort mit einem kurzen Beweis.

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)	n	$\in \Delta(1+(1+(-1)^n)n)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(f)	$\sqrt{n}$	$\in \Delta(\ln n)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.
(g)	n^2	$\in \Delta(1+\lvert tan(n)\rvert)$	☐ o	☐ $\mathcal{O}$	☐ ω	☐ Ω	☐ Θ	☐ u.

Bsp. $n \in \Delta(n^2)$

$f(n) = n \quad g(n) = n^2$

$$\Rightarrow \lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \lim_{n\to\infty}\left(\frac{\cancel{n}}{n^{\cancel{2}}}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right) = 0$$

- $f(n)$ wird im Unendlichen „unendlich klein" im Vergleich zu $g(n)$.
- Dies kann durch keinen konstanten Faktor geändert werden

$\Rightarrow f(n) \in o(g(n))$

Aufgabe 2.3

(a) $7n^3 \in \Delta(3n^7)$ ☒ o ☐ $\mathcal{O}$ ☐ ω ☐ Ω ☐ Θ ☐ u.

- Verwendung der Limes-Definition

$$\lim_{n\to\infty}\left(\frac{7n^3}{3n^7}\right) = \lim_{n\to\infty}\left(\frac{7}{3n^4}\right) = 0$$

$$\Rightarrow 7n^3 \in o(3n^7)$$

(b) $\sqrt[3]{\sqrt{3n}} \in \Delta\left(\sqrt{\sqrt[3]{2n}}\right)$ ☐ o ☐ $\mathcal{O}$ ☐ ω ☐ Ω ☒ Θ ☐ u.

(1) Wurzeln können als Potenzen vereinfacht werden:

- $\sqrt[3]{\sqrt{3n}} = \left((3n)^{\frac{1}{2}}\right)^{\frac{1}{3}} = (3n)^{\frac{1}{6}}$
- $\sqrt{\sqrt[3]{2n}} = \left((2n)^{\frac{1}{2}}\right)^{\frac{1}{3}} = (2n)^{\frac{1}{6}}$

(2) Limes-Definition verwenden:

$$\lim_{n\to\infty}\left(\frac{(3n)^{\frac{1}{6}}}{(2n)^{\frac{1}{6}}}\right) = \lim_{n\to\infty}\left(\left(\frac{3n}{2n}\right)^{\frac{1}{6}}\right) = \lim_{n\to\infty}\left(\left(\frac{3}{2}\right)^{\frac{1}{6}}\right) = \sqrt[6]{\frac{3}{2}}$$

- Im Unendlichen trennt beide Funktionen der konstante Faktor $c = \sqrt[6]{\frac{3}{2}}$
Somit gibt es ein n ab dem ein Faktor $d < c$ die eine Funktion kleiner werden lässt. (Analog mit $e > c$ um größer zu werden)

$$\Rightarrow \sqrt[3]{\sqrt{3n}} \in \Theta\left(\sqrt{\sqrt[3]{2n}}\right)$$

Aufgabe 2.3

(c) $n! \in \Delta(4^n)$ ☐ o ☐ $\mathcal{O}$ ☒ ω ☐ Ω ☐ Θ ☐ u.

Variante Musterlösung:

- Zeige ω für eine Funktion die in jedem Punkt kleiner ist als $n!$
- Daraus folgt ω auch für $n!$ (da diese Funktion größer ist)
- Nutze Aussage aus 1.5 (e): $n! \geq n^{\frac{n}{2}} = \sqrt{n}^{\,n}$

$\Rightarrow n! \in \Omega(\sqrt{n}^{\,n})$

Abschätzung: $\sqrt{n} \geq 16$ (im Grenzwert)

$$\lim_{n\to\infty}\left(\frac{\sqrt{n}^{\,n}}{4^n}\right) = \lim_{n\to\infty}\left(\left(\frac{\sqrt{n}}{4}\right)^n\right) \geq \lim_{n\to\infty}\left(\left(\frac{16}{4}\right)^n\right) = \infty$$

$\Rightarrow \sqrt{n}^{\,n} \in \omega(4^n)$

- $n! \in \Omega(\sqrt{n}^{\,n}) \wedge \sqrt{n}^{\,n} \in \omega(4^n) \Rightarrow n! \in \omega(4^n)$

Transitivitätsregeln der O-Notation (2.4 (p))

(d) $n \in \Delta((2 + (-1)^n)n)$ ☐ o ☐ $\mathcal{O}$ ☐ ω ☐ Ω ☒ Θ ☐ u.

- Stückweise definierbare Funktion

$$(2+(-1)^n)\cdot n = \begin{cases} 3n & \text{für } n \text{ gerade} \\ n & \text{für } n \text{ ungerade} \end{cases}$$

- Hier relativ offensichtlich:

Für Faktor $c=1$ gilt $n \leq 1 \cdot 3n$ und $n \leq 1 \cdot n$ $\Rightarrow n \in \mathcal{O}((2+(-1)^n)n)$

$c=\frac{1}{3}$ gilt $n \geq \frac{1}{3} \cdot 3n$ und $n \geq \frac{1}{3} n$ $\Rightarrow n \in \Omega((2+(-1)^n)n)$

$\Rightarrow n \in \Theta((2+(-1)^n)n)$

- Ansonsten: evtl. Limes-Definition für beide Teile der stückweisen Funktion überprüfen

Aufgabe 2.3

(e) $n \in \Delta(1+(1+(-1)^n)n)$ ☐ o ☐ $\mathcal{O}$ ☐ ω ☒ Ω ☐ Θ ☐ u.

$$1+(1+(-1)^n)n = \begin{cases} 1+2n & \text{für } n \text{ gerade} \\ 1 & \text{für } n \text{ ungerade} \end{cases}$$

- Für Faktor z.B. $c = \frac{1}{5}$ gilt: $n \geq \frac{1}{5} + \frac{2}{5}n$ $(\forall n \geq 1)$

  $n \geq \frac{1}{5}$

$\Rightarrow n \in \Omega(1+(1+(-1)^n)n)$ *

---

* Achtung: $\omega(\ldots)$ wäre auch noch möglich!

Für Faktor z.B. $c=1$ gilt: $n \leq 1+2n$ $(\forall n \geq 1)$ für $\omega(\ldots)$ scheitert

$\Rightarrow$ kein möglicher Startwert n_0, Bedingung an Faktor $c=1$ (gilt nicht $\forall c > 0$)

(f) $\sqrt{n} \in \Delta(\ln n)$ ☐ o ☐ $\mathcal{O}$ ☒ ω ☐ Ω ☐ Θ ☐ u.

$$\lim_{n\to\infty}\left(\frac{n^{\frac{1}{2}}}{\ln(n)}\right) = \frac{\infty}{\infty} \longrightarrow \lim_{n\to\infty}\left(\frac{n^{\frac{1}{2}}}{\ln(n)}\right) = \lim_{n\to\infty}\left(\frac{\frac{d}{dn}\left(n^{\frac{1}{2}}\right)}{\frac{d}{dn}\ln(n)}\right) = \lim_{n\to\infty}\left(\frac{\frac{1}{2}\cdot n^{-\frac{1}{2}}}{\frac{1}{n}}\right) = \lim_{n\to\infty}\left(\frac{1}{2}\sqrt{n}\right) = \infty$$

Verwende Regel von de L'Hospital:

Wenn $\lim_{n\to\infty}\left(\frac{f(x)}{g(x)}\right) = \frac{\infty}{\infty}$

dann gilt $\lim_{n\to\infty}\left(\frac{f(x)}{g(x)}\right) = \lim_{n\to\infty}\left(\frac{f'(x)}{g'(x)}\right)$ sofern dieser Grenzwert existiert

$\Rightarrow \sqrt{n} \in \omega(\ln(n))$

Aufgabe 2.3

(g) $n^2 \in \Delta(1 + |tan(n)|)$ $\square\ o$ $\square\ \mathcal{O}$ $\square\ \omega$ $\square\ \Omega$ $\square\ \Theta$ $\boxtimes$ u.

- Da $\tan(n)$ periodisch ist, macht der Grenzwert für $n \to \infty$ in diesem Fall keinen Sinn

↳ Betrachte Werte von $1 + |\tan(n)|$: Schwankt innerhalb einer Periode von ∞ zu 1, dann wieder nach ∞

⇒ Innerhalb einer Periode wird $1 + |\tan(n)|$ größer als n^2 starten, irgendwann $\underbrace{\text{kleiner gleich } n^2}_{\text{für } n \text{ groß genug}}$ sein und später wieder größer werden (egal welcher positive Faktor verwendet wird)

⇒ Weder $\mathcal{O}$ noch Ω möglich (somit auch nicht Θ oder die stärkeren Aussagen von o und ω)

↳ unbestimmt

Aufgabe 2.4
$\mathcal{O}$h ja, noch mehr Wachstum

Aufgabe 2.4

Seien $f, g, h : \mathbb{N}_0 \to \mathbb{R}$ Funktionen mit $\exists n_0 : \forall n > n_0 : f(n), g(n), h(n) > 0$. Zeigen Sie die folgenden „Transitivitätsregeln“.

$$
\begin{array}{llllll}
\text{(a)} & f(n) \in o(g(n)) & & & \Rightarrow & f(n) \in \mathcal{O}(g(n)) \\
\text{(b)} & f(n) \in \mathcal{O}(g(n)) & \text{und} & g(n) \in \mathcal{O}(h(n)) & \Rightarrow & f(n) \in \mathcal{O}(h(n)) \\
\text{(c)} & f(n) \in o(g(n)) & \text{und} & g(n) \in \mathcal{O}(h(n)) & \Rightarrow & f(n) \in o(h(n))
\end{array}
$$

Hinweis: Solche Regeln sind nützlich wenn man Beweise für komplexe Funktionen durchführen muss, welche in einfachere aufgeteilt werden können.

Aufgabe 2.4

(a) $f(n) \in o(g(n)) \quad \Rightarrow \quad f(n) \in \mathcal{O}(g(n))$

Annahme: $\forall C > 0 : \exists n_0 > 0 : \forall n \geq n_0 : f(n) \leq C \cdot g(n)$

"Für jeden positiven Faktor C gibt es einen Startwert n_0 ab dem $f(n) \leq C \cdot g(n)$ immer gilt"

Zu zeigen: $f(n) \in \mathcal{O}(g(n)) \iff \exists c > 0 : \exists n_0 > 0 : \forall n \geq n_0 : f(n) \leq c \cdot g(n)$

"Es gibt (mindestens) einen positiven Faktor c und einen zugehörigen Startwert n_0 ab dem $f(n) \leq c \cdot g(n)$ immer gilt"

$\Rightarrow$ Trivial: Annahme erfüllt dies bereits für jeden beliebigen positiven Faktor

↳ Wähle beliebigen Faktor und zugehörigen Startwert.

Aufgabe 2.4

(b) $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in \mathcal{O}(h(n)) \Rightarrow f(n) \in \mathcal{O}(h(n))$

Annahmen: (1) $\exists c_1 > 0 : \exists n_1 > 0 : \forall n \geq n_1 : f(n) \leq c_1 \cdot g(n)$

(2) $\exists c_2 > 0 : \exists n_2 > 0 : \forall n \geq n_2 : g(n) \leq c_2 \cdot h(n)$

Zu zeigen: (3) $\exists c_3 > 0 : \exists n_3 > 0 : \forall n \geq n_3 : f(n) \leq c_3 \cdot h(n)$

- Wähle $n_3 = \max\{n_1, n_2\}$: Somit gelten (1) und (2) noch mit n_3 als Startwert

↳ $\forall n \geq n_3 : f(n) \leq c_1 \cdot g(n) \leq c_1 \cdot c_2 \cdot h(n)$ (Einsetzen von (2) in (1))

$\Rightarrow f(n) \leq (c_1 \cdot c_2) \cdot h(n)$

$\Rightarrow$ Die Existenz von $c_3 = c_1 \cdot c_2$ und $n_3 = \max\{n_1, n_2\}$ impliziert somit Aussage (3)

Aufgabe 2.4

(c) $f(n) \in o(g(n))$ und $g(n) \in \mathcal{O}(h(n)) \Rightarrow f(n) \in o(h(n))$

(1) $\forall C_1 > 0 : \exists n_1 > 0 : \forall n \geq n_1 : f(n) \leq C_1 \cdot g(n)$

(2) $\exists c_2 > 0 : \exists n_2 > 0 : \forall n \geq n_2 : g(n) \leq c_2 \cdot h(n)$

Zu zeigen: (3) $\forall C_3 > 0 : \exists n_3 > 0 : \forall n \geq n_3 : f(n) \leq C_3 \cdot h(n)$

- Sei $C_1 > 0$ beliebig aber <u>fixiert</u>
- Wähle $n_3 = \max\{n_1, n_2\}$: somit gelten (1) und (2) auch für den Startwert n_3
- Somit gilt $f(n) \leq C_1 \cdot g(n) \leq \underbrace{C_1 \cdot c_2}_{C_3} \cdot h(n)$
- Setzt man nun $C_3 = C_1 \cdot c_2$ und $n_3 = \max\{n_1, n_2\}$ impliziert dies (3)

Beachte: C_3 kann jeden beliebigen Wert größer 0 annehmen da dies auch für C_1 möglich ist $\Rightarrow$ setze $C_1 = C_3 / c_2$ für beliebiges C_3