

Einzelprüfung „Theoretische Informatik / Algorithmen (vertieft)“

Einzelprüfungsnummer 66115 / 2020 / Herbst

## Thema 2 / Teilaufgabe 2 / Aufgabe 2

(Beweisen von Aussagen)

**Stichwörter:** Komplexitätstheorie

Beweisen Sie die folgenden Aussagen:

(a) Sei  $f(n) = 2 \cdot n^2 + 3 \cdot n + 4 \cdot (\log n) + 7$ . Dann gilt  $f \in O(n^2)$ .

(b) Sei  $f(n) = 4^n$ . Dann gilt nicht  $f \in O(2^n)$ .

Lösungsvorschlag

(c) Sei  $f(n) = (n+1)!$  (d. h. die Fakultät von  $n+1$ ). Dann gilt  $f \in O(n^n)$ .

Lösungsvorschlag

z. B.  $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 \cdot 5^5$

(d) Sei  $f: \mathbb{N} \leftarrow \mathbb{N}$  definiert durch die folgende Rekursionsgleichung:

$$f(n) = \begin{cases} 3, & \text{für } n = 1 \\ (n-1)^2 + f(n-1), & \text{für } n > 1 \end{cases}$$

Dann gilt  $f \in O(n^3)$

Lösungsvorschlag

$$\begin{aligned} f(n) &= (n-1)^2 + f(n-1) + \dots + f(1) \\ &= (n-1)^2 + (n-1)^2 + f(n-2) + \dots + f(1) \\ &= \underbrace{(n-1)^2 + \dots + (n-1)^2}_{n} + 3 \\ &= \underbrace{(n-1)^2 + \dots + (n-1)^2}_{n-1} + 3 \\ &= (n-1)^2 \cdot (n-1) + 3 \\ &= (n-1)^3 + 3 \end{aligned}$$



## Die Bschlangaul-Sammlung

### Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66115/2020/09/Thema-2/Teilaufgabe-2/Aufgabe-2.tex>