

Algorithmen und Lernverfahren

Kay Förster

29. Oktober 2015

1. Berechnen Sie die Laufzeit des folgenden Algorithmuses:

$$\sum_{m=0}^k a_m n^m \text{ mit } a_k > 0, a_m \geq 0$$

Lösung:

$$\begin{aligned} LZ &\leq \sum_{m=0}^k a_m n^m \\ &\leq a_0 n^0 + \dots + a_k n^k \\ &\leq a_k n^k \in \mathcal{O}(n^k) \end{aligned}$$

2. Zeigen Sie:

- (a) $\log(n!) \in \mathcal{O}(n \log n)$
- (b) $\binom{n}{k} \in \mathcal{O}(n^k)$
- (c) $2^{n+\mathcal{O}(1)} \in \mathcal{O}(2^n)$

Lösung:

(a)

$$\begin{aligned} \mathcal{O}(\log(n!)) &\subseteq \mathcal{O}(n \log n) \\ \mathcal{O}(\log(n \cdot n - 1 \cdot \dots \cdot n - (n-1))) &\subseteq \\ \mathcal{O}(\log(n^n)) &\subseteq \end{aligned}$$

(b)

(c)

$$2^{n+\mathcal{O}(1)} \subseteq 2^{n+c} = 2^n \cdot 2^c \in \mathcal{O}(2^n)$$

3. Zeigen Sie das $\frac{n(n+1)}{2} \in \Theta(n^2)$ gilt.

Lösung:

$$\begin{aligned}\mathcal{O}\left(\frac{n(n+1)}{2}\right) &= \mathcal{O}(n(n+1)) = \mathcal{O}(n) \cdot \mathcal{O}(n+1) = \mathcal{O}(n) \cdot \mathcal{O}(n) = \mathcal{O}(n^2) \\ \frac{n(n+1)}{2} &\geq \frac{n^2}{2} \in \Theta(n^2) \\ \hookrightarrow \mathcal{O}(n^2) \cap \Omega(n^2) &= \Theta(n^2)\end{aligned}$$

4. Zeigen Sie ob $n \in \Theta(2^n)$ gilt.

Lösung:

$$n \in \Theta(2^n) = \mathcal{O}(2^n) \cap \underbrace{\Omega(2^n)}_{\text{Verletzt}}$$

2^n wächst schneller als n , somit kann die untere Schranke nicht eingehalten werden.