

Algorithmen & Datenstrukturen

Übungsstunde 3

Miniquiz 1

- Passwort: **subarray1**
- 5 Fragen
- 6 Minuten
- 0.2 Bonus Punkte pro richtige Antwort

Programm

- Miniquiz
- Rückmeldung Serie 1
- Besprechung Aufgaben
- Theory Recap
- Prüfungsaufgabe HS19 T1 a) FS19 T1 e) (Kahoot!) HS19 T2 a)
- Code Demo MSS

Rückmeldung Serie 1

- Gut gelöst
- Bei Induktionshypothese wichtig, dass wir genau ein k fixieren
- Angeben wo Induktionshypothese benutzt wurde
- Bei 1.3 Rechnung zu ende führen
- Bitte kein rot verwenden

Besprechung Miniquiz

$$n^8 - 800n^7 - 10000n^6 + 200n^2 \leq O(0.25n^7)$$

- ☐ Wahr
- ☐ Falsch

Let $f(n) = 4e^n$, for which of the following definitions of $g(n)$ is $f(n) \leq O(g(n))$?

True	False		
☐	☐	$g(n) = n^4 + n^2$	
☐	☐	$g(n) = n^{100} + e^{100 \ln(n)}$	
☐	☐	$g(n) = 2^{3n}$	

Bewertung: **Teilpunkte** 

You want to show using induction that a statement $A(n)$ holds for all n such that $n \geq 2$.

Which of the following combinations of *base case* and *induction step* would form a valid proof?

Wählen Sie eine oder mehrere Antworten:

☐ a. *Base case:* $A(2)$ holds.

Induction step: $A(k) \implies A(k+1)$ for all integers $k \geq 2$

☐ b. *Base case:* $A(2)$ holds.

Induction step: $A(k) \implies A(k+2)$ for all integers $k \geq 2$.

☐ c. *Base case:* $A(2), A(3)$ holds.

Induction step: $A(k) \implies A(k+2)$ for all integers $k \geq 2$.

$$\sum_{i=1}^n i^2 \leq O(n^2 \log(n))?$$

☐ Wahr

☐ Falsch

Consider the following recursive formula:

$$T(1) = 0$$

$$T(n) = 2T(n/2) + 5n$$

Assume n is of the form $n = 2^k$ for $k \in \{0, 1, 2, 3, \dots\}$.

Is $T(n) \leq O(n)$?

☐ Wahr

☐ Falsch

Theory Recap

Inhalt

- Maximum Subarray Sum
- Input Pseudo Code
- Theorem von Aufgabenblatt

Maximum Subarray Sum

- naiv: Alle Teilsummen berechnen $\Theta(n^3)$
- immernoch naiv: Alle Teilsummen mit vorheriger Teilsumme $\Theta(n^2)$
- bisschen besser: Rekursiv $\Theta(n \cdot \log(n))$
- am besten: Induktiv $\Theta(n)$

Maximum Subarray Sum

Beste Lösung

j	0	1	2	3	4	5	6	7	8	9
a_j	5	8	-4	1	-12	4	13	-2	-4	-1
R_j	5	13	9	10	-2	4	17	15	11	10
R_M	5	13	13	13	13	13	17	17	17	17

$$R_j = \begin{cases} R_{j-1} + a_j & \text{wenn } R_{j-1} \geq 0 \\ a_j & \text{sonst} \end{cases}$$

Pseudo-Code

Algo1($A[]$)

curr $\leftarrow$ 0

res $\leftarrow$ 0

for $i = 0 \dots n-1$ do

if curr $\geq$ 0 do

curr $\leftarrow$ curr + $A[i]$

end

else do

curr $\leftarrow$ $A[i]$

end

res $\leftarrow$ Max(curr, res)

end

return res

$A[] \leftarrow$ Array of length n

Theorem Aufgabenblatt 3

- If $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$, then $f \leq O(g)$, but $f \neq \Theta(g)$
- If $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = C \in \mathbb{R}^+$, then $f = \Theta(g)$
- If $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$, then $f \geq \Omega(g)$, but $f \neq \Theta(g)$

Prüfungsaufgaben

Kahoot! O-Notation Quiz



Prüfungsaufgaben

Runtime Snippets

Algorithm 1

```
for  $j = 1, \dots, n$  do  
    for  $k = j, \dots, n$  do  
         $f()$ 
```

Algorithm 2

```
for  $j = 1, \dots, n$  do  
    for  $k = 1, 2, 3, 4, 5, \dots, 2^j$  do  
        for  $l = 1, \dots, 42$  do  
             $f()$ 
```

Leetcode of the Week

- 121. Best Time to Buy and Sell Stock
- leetcode.com/problems/best-time-to-buy-and-sell-stock

Peergrading

- Aufgabe 2.4
- Grading Scheme in Lösungen beachten