QUIZ-NACHBESPRECHUNG

1) You want to show using induction that a statement $A(n)$ holds for all n of the form $n = 2^k$, with $k \in \mathbb{N}_0 = \{0, 1, 2, 3, \ldots\}$

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

X *Base case:* $A(1)$ holds.
Induction step: $A(n) \implies A(2n)$ for all integers $n \geq 1$.

Base case: $A(0)$ holds.
Induction step: $A(2^k) \implies A(2^{k+1})$ for all integers $k \geq 1$.

Base case: $A(1)$ holds.
Induction step: $A(2^k) \implies A(2^{k+1})$ for all integers $k \geq 1$.

X *Base case:* $A(1)$ holds.
Induction step: $A(2^k) \implies A(2^{k+1})$ for all integers $k \geq 0$.

2) $\sum_{k=1}^{n} k^{0.3} \leq O(n^{1.3})$? True

3) Recall that $f = \Theta(g)$ if and only if $f \leq O(g)$ and $g \leq O(f)$.

Is it true that $n^2 + n - 1 = \Theta(n^3 - n^2)$?

False

4) Consider the following pseudocode snippet:

$i \leftarrow 1$
while $i \leq n$:
    $i \leftarrow i + 2$
    $f()$

$\# \text{calls} = \sum_{i=1}^{\frac{n}{2}} 1 = \frac{n}{2} = \Theta(\frac{n}{2}) = \Theta(n)$

Which of the following expressions in Θ-notation correctly describe the number of calls to f?

$\Theta(n^2)$ $\Theta(n \log n)$ $\underline{\Theta(n)}$ $\underline{\Theta(\frac{n}{2})}$

5) Consider the following pseudocode snippet:

$x \leftarrow n^2$
while $x \geq 2$:
    $x \leftarrow x/2$ (x is not necessarily an integer!)
    $f()$

Assume $n \geq 2$. Which of the following expressions in Θ-notation correctly describes the number of calls to f?

$\underline{\Theta(\log_2 n)}$ $\Theta(n^2 \log_2 n)$ $\Theta(\frac{n}{2})$ $\Theta((\log_2 n)^2)$

SERIE 1 - COMMON MISTAKES

- Aufbau eines Induktionsbeweises:

  1. Base Case für <u>kleinstmögliches</u> n
  2. ausformulierte Induktionshypothese
  3. Induktionsschritt
  4. Abschlussatz → was habt ihr gerade bewiesen?

  → forward proofs sind good practice, backward proofs (meist) nicht

- Bei jeder Teilaufgabe einen abschliessenden Satz schreiben!

- Bitte kein Rot benutzen

- Erinnerung an Abgabeformat:

  Theorie1_Kürzel1_Kürzel2
  PG1_Kürzel1_Kürzel2

NACHBESPRECHUNG SERIE 2

Aufgabe 1.2b)

Let x be any real number. Prove via mathematical induction that for every positive integer n, we have

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i ,$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} .$$

We use a standard convention $0! = 1$, so $\binom{n}{0} = \binom{n}{n} = 1$ for every positive integer n.

In your solution, you should address the base case, the induction hypothesis and the induction step.

Hint: You can use the following fact without proof: for every $1 \le i \le n$,

$$\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i} .$$

Base Case $n=1$: $(1+x)^1 = \sum_{i=0}^{1} \binom{1}{i} x^i = \underbrace{\binom{1}{0} x^0 + \binom{1}{1} x^1}_{=1+x}$

I.H.: ...

I.S.: $(1+x)^{k+1} = (1+x)(1+x)^k \overset{\text{I.H.}}{=} (1+x) \cdot \sum_{i=0}^{k} \binom{k}{i} x^i$

$= \sum_{i=0}^{k} \binom{k}{i} x^i + \sum_{i=0}^{k} \binom{k}{i} x^{i+1}$

$= \binom{k}{0} \cdot x^0 + \sum_{i=1}^{k} \binom{k}{i} x^i + \sum_{i=1}^{k} \binom{k}{i-1} x^i + \binom{k}{k} \cdot x^{k+1}$

$\binom{k}{0} = \binom{k+1}{0} = \binom{k}{k} = \binom{k+1}{k+1} = 1$

$= \underbrace{\binom{k}{0}}_{=\binom{k+1}{0}} \cdot x^0 + \sum_{i=1}^{k} \binom{k+1}{i} x^i + \underbrace{\binom{k}{k}}_{=\binom{k+1}{k+1}} \cdot x^{k+1}$

$\sum_{i=1}^{k} \binom{k}{i} x^i + \binom{k}{i-1} x^i$

$= \sum_{i=1}^{k} \Big(\underbrace{\binom{k}{i} + \binom{k}{i-1}}_{=\binom{k+1}{i}}\Big) \cdot x^i$

$= \sum_{i=0}^{k+1} \binom{k+1}{i} x^i$

+ conclusion

Aufgabe 2.3

Let $f : \mathbb{N} \to \mathbb{R}_+$ be a function, with $f(n) \le O(n)$. A colleague tried to prove that $e^{f(n)} \le O(e^n)$. You found their notes, in which they start with the statement they want to show, and derive a series of equivalent statements. The notes read:

		justifications for "$\iff$":	
	$e^{f(n)} \le O(e^n)$		
$\iff$	$e^{f(n)} \le C \cdot e^n$, for some $C > 0$	Use Definition 1 on the first page	(1)
$\iff$	$\log\left(e^{f(n)}\right) \le \log\left(C \cdot e^n\right)$	Take the log on both sides	(2)
$\iff$	$f(n) \le \log C + n$	$\log(C \cdot e^n) = \log C + n$	(3)
$\iff$	$f(n) \le O(n)$	$n + \log C \le O(n)$	(4)

a) $2n$

$f(n) \le O(n) \not\Rightarrow f(n) \le \log C + n$

Aufgabe 2.5

(a) Show that $\ln(n!) \le O(n \ln n)$.

Hint: *You can use the fact that* $n! \le n^n$ *for* $n \in \mathbb{N}_{\ge 2}$ *without proof.*

(b) Show that $n \ln n \le O(\ln(n!))$.

Hint: *You can use the fact that* $\left(\frac{n}{2}\right)^{\frac{n}{2}} \le n!$ *for* $n \in \mathbb{N}_{\ge 2}$ *without proof.*

a)

$\ln(n!) \le \ln(n^n) = n \cdot \ln(n) \le O(n \ln n)$

b) $\ln\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right) = \frac{n}{2} \ln\left(\frac{n}{2}\right) \le \ln(n!)$

$= \frac{n}{2}(\ln(n) - \ln(2))$

$\lim_{n\to\infty} \frac{n \ln n}{\frac{n}{2}(\ln(n) - \ln(2))} = \lim_{n\to\infty} \frac{\ln(n)}{\frac{1}{2}(\ln(n) - \ln(2))} = \lim_{n\to\infty} \frac{1}{\frac{1}{2} - \underbrace{\frac{\ln(2)}{\ln(n)}}_{0}} = 2$

$n \ln(n) \le O\left(\frac{n}{2}\ln(n) - \frac{n}{2}\ln(2)\right)$

MAXIMUM SUBARRAY SUM (MSS)

Problem: gegeben: Array von n Zahlen $a_1, \dots, a_n \in \mathbb{R}$
gesucht: Teilarray mit maximaler Summe

MSS-NAIV$(a_1, \dots, a_n)$

---

maxS $\leftarrow 0$
Für $i \leftarrow 1, \dots, n$ (alle Anfänge)
  Für $j \leftarrow i, \dots, n$ (alle Enden)
    $S \leftarrow \sum_{k=i}^{j} a_k$ (berechne Summe)
    Merke maximales S

$\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=i}^{j} 1 = \sum_{i=1}^{n}\sum_{j=1}^{n} (j-i+1)$

Laufzeit: $\Theta(n^3)$

möglicher Java-Code:

```
double maximalesS=0.0;
for (int i=0; i<n; i=i+1)
    for (int j=i; j<n; j=j+1) {
        double S=0.0;
        for (int k=i; k<=j; k=k+1) {
            S=S+a[k];
            if (S>maximalesS)
                maximalesS=S;
        }
    }
```

Idee: Nutzen von Präfixsummen

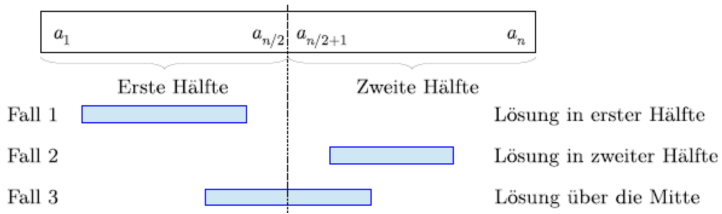
MSS-PRÄFIXSUMMEN$(a_1, \dots, a_n)$

$S_0 \leftarrow 0$
Für $i \leftarrow 1, \dots, n$
  $S_i \leftarrow S_{i-1} + a_i$
maxS $\leftarrow 0$
Für $i \leftarrow 1, \dots, n$
  Für $j \leftarrow i, \dots, n$
    $S \leftarrow S_j - S_{i-1}$ $\leftarrow O(1)$
    Merke maximales S

Laufzeitabschätzung:
→ Berechnung der Präfixsummen in $\Theta(n)$
→ Anzahl der Teilabschnitte $\frac{n(n-1)}{2} = \binom{n}{2} = \frac{n!}{(n-2)! \cdot 2!}$
→ Gesamtlaufzeit: $\Theta(n^2)$

Idee: Divide-and-Conquer

Aufteilen des Arrays in zwei Teilarrays
→ Fall 1: Lösung liegt vollständig im ersten Teil
→ Fall 2: Lösung liegt vollständig im zweiten Teil
→ Fall 3: Lösung läuft über die Mitte



MSS-DIVIDE-AND-CONQUER$(a_1, \dots, a_n)$

Wenn $n = 1$ ist, dann gib $\max\{a_1, 0\}$ zurück.
Wenn $n > 1$ ist:
  Teile die Eingabe in $A_1 = \langle a_1, \dots, a_{n/2}\rangle$ und $A_2 = \langle a_{n/2+1}, \dots, a_n\rangle$ auf.
  Berechne rekursiv den Wert W_1 einer besten Lösung für das Array A_1.
  Berechne rekursiv den Wert W_2 einer besten Lösung für das Array A_2.
  Berechne grösste Suffixsumme S in A_1.
  Berechne grösste Präfixsumme P in A_2.
  Setze $W_3 \leftarrow S + P$.
  Gib $\max\{W_1, W_2, W_3\}$ zurück.

→ $\Theta(n \log n)$

geht es besser?

JA!

MSS-INDUKTIV$(a_1, \ldots, a_n)$

```
randmax ← 0
maxS ← 0
Für i ← 1, ..., n:
    randmax ← randmax + a_i
    Wenn randmax > maxS:
        maxS ← randmax
    Wenn randmax < 0:
        randmax ← 0
Gib maxS zurück.
```

Laufzeit: $\Theta(n)$

Beispiel:

	2	3	-6	7	-5	12	4	-2	8	-20
Maximum	2	5	5	7	7	14	18	18	26	

O-NOTATION (ERGÄNZUNG)

Recap letzte Woche

Definition: Sei $f: \mathbb{N} \to \mathbb{R}^+$

$$O(f) := \{g: \mathbb{N} \to \mathbb{R}^+ \mid \exists c>0\, \exists n_0 \in \mathbb{N}\, \forall n \geq n_0: g(n) \leq c \cdot f(n)\}$$

$\uparrow$
menge aller Funktionen,
die nicht schneller wachsen als f

Diese Woche

Für $f: N \to \mathbb{R}$

$$\Omega(f) := \{g: N \to \mathbb{R}_+ \mid f \leq O(g)\}$$

$$\Theta(f) := \{g: N \to \mathbb{R}_+ \mid g \leq O(f) \wedge f \leq O(g)\}$$

Theorem 1. *Let N be an infinite subset of $\mathbb{N}$ and $f: N \to \mathbb{R}_+$ and $g: N \to \mathbb{R}_+$.*

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

Sandwich-Theorem

$f: \mathbb{N} \to \mathbb{R}$, $g: \mathbb{N} \to \mathbb{R}$, $h: \mathbb{N} \to \mathbb{R}$

$g(n) \leq f(n) \leq h(n)$ für alle $n \in \mathbb{N}$ $\Rightarrow$ $\lim_{n\to\infty} f(n) = c$

$\lim_{n\to\infty} g(n) = c$ und $\lim_{n\to\infty} h(n) = c$

Anwendungsbeispiele:

a) Zeige $(\sin(n)+2)n = \Theta(n)$ $\qquad -1 \leq \sin(n) \leq 1 \quad \forall n \in \mathbb{N}$

$$\Theta(n) = n \leq (\sin(n)+2)n \leq 3n = \Theta(n)$$

b) Beweise $\sum_{i=1}^{n} \sum_{j=1}^{i} j = \Theta(n^3)$ $\qquad \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

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

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

Aufgabentyp: Laufzeiten von Code Snippets bestimmen

Algorithm 1

```
i ← 0
while i ≤ n do
    f()
    f()
    i ← i + 1
j ← 0
while j ≤ 2n do
    f()
    j ← j + 1
```

$\leftarrow 2$

$\leftarrow 1$

$$\#\text{calls} = \sum_{i=0}^{n} 2 + \sum_{j=0}^{2n} 1 = (n+1)\cdot 2 + (2n+1) = 4n+3 = \Theta(n)$$

Algorithm 2

```
for i = 1, ..., n do
    k ← 1
    while k ≤ i² do
        f()
        k ← 2k
    f()
```

$$\#\text{calls} = \sum_{i=1}^{n}\left(\left(\sum_{k=1}^{\log_2(i^2)} 1\right)+1\right) = \sum_{i=1}^{n}\sum_{k=1}^{\log_2(i^2)} 1 + \sum_{i=1}^{n} 1 = \sum_{i=1}^{n} \log_2(i^2) + n$$

$$= 2\cdot \underbrace{\sum_{i=1}^{n} \log_2(i)}_{\log_2(1)+\log_2(2)+\ldots+\log_2(n) = \log_2(1\cdot 2\cdot\ldots\cdot n) = \log_2(n!)} + n = 2\log_2(n!) + n \leq 2\cdot \log_2(n^n) + n = 2n\log_2(n) + n$$

$$= \Theta(n\log n)$$

$\geq 2\cdot \log_2\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right)$

$\log(a) + \log(b) = \log(a\cdot b)$

FS23

Asymptotic notation quiz: For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

5/60 P. ⇒ etwa 10 Minuten in der Prüfung Zeit

Assume $n \geq 4$.

Claim	true	false
$n^3 + n^4 = \Theta(n^4)$	☒	☐
$n^{10} \leq O(\log(n)^{100})$	☐	☒
$1\cdot 2\cdot 3\cdot\ldots\cdot n \leq O(2^n)$	☐	☒
Suppose $a_1 = 1$ and $a_{i+1} = 3a_i + 1$ for all $i \geq 2$. Then $a_n \leq O(4^n)$.	☒	☐
$2^{10\log(n)} = \Theta(2^{20\log(n)})$ $= (2^{10\log(n)})^2$	☐	☒

$n! \geq \left(\frac{n}{2}\right)^{\frac{n}{2}} = 2^{\log_2\left(\frac{n}{2}\right)\cdot\frac{n}{2}}$

geom. Summe: $S_n = \sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$

$a_1 = 1 \quad a_2 = 4 \quad a_3 = 13 \quad a_4 = 40$

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