Einführung in die Programmierung

Algorithmen-Tutorium

Wintersemester 2025/2026

Emilio Pielsticker

Arbeitsgruppe Systemsoftware
Angewandte Informatik 12

technische universität
dortmund

Arbeitsgruppe
Systemsoftware

Thema Heute: **Algorithmen**

Technische Umsetzung von Algorithmen durch **Programmierung** in **C**

Später auch in **C++**

Ein **Algorithmus** ist eine endliche Folge von wohldefinierten, ausführbaren Anweisungen zur Lösung eines Problems

- **Problem**: Berechnung einer totalen Abbildung $\mathrm{f} : D \subseteq \mathbb{N} \to \mathbb{N}$
- Ein Algorithmus heißt genau dann **korrekt**, wenn er bei jeder **zulässigen** Eingabe (dh. $x \in D$) in endlicher Zeit das Bild $\mathrm{f}(x)$ liefert
- Bei einem **deterministischen** Algorithmus ist die Folge der auszuführenden Anweisungen für jede beliebige Eingabe exakt vorhersagbar

- Der Gauß-Jordan-**Algorithmus** löst lineare Gleichungssysteme durch geschickte Elimination innerhalb der Koeffizientenmatrix
- Zulässige Eingabe: Koeffizientenmatrix $[\mathrm{A}\,;\vec{b}]$ eines lösbaren LGS
- Man wähle ein **Pivot-Element** (i,i) aus der Hauptdiagonalen
- Die i-te Zeile sei derart zu skalieren, dass das Pivot-Element zu 1 wird
- Von allen anderen Zeilen $j \neq i$ sei ein Vielfaches der i-ten Zeile so zu subtrahieren, dass diese darauf an der i-ten Stelle eine Null enthalten
- Wiederhole die Schritte für alle n Pivot-Elemente
- Ausgabe des Algorithmus: $[\mathbb{1}\,;\vec{x}]$ mit $\vec{x}$ als **Lösungsvektor**

$$\mathrm{A} \in \mathbb{R}^{n\times n}, \vec{b} \in \mathbb{R}^{n}$$

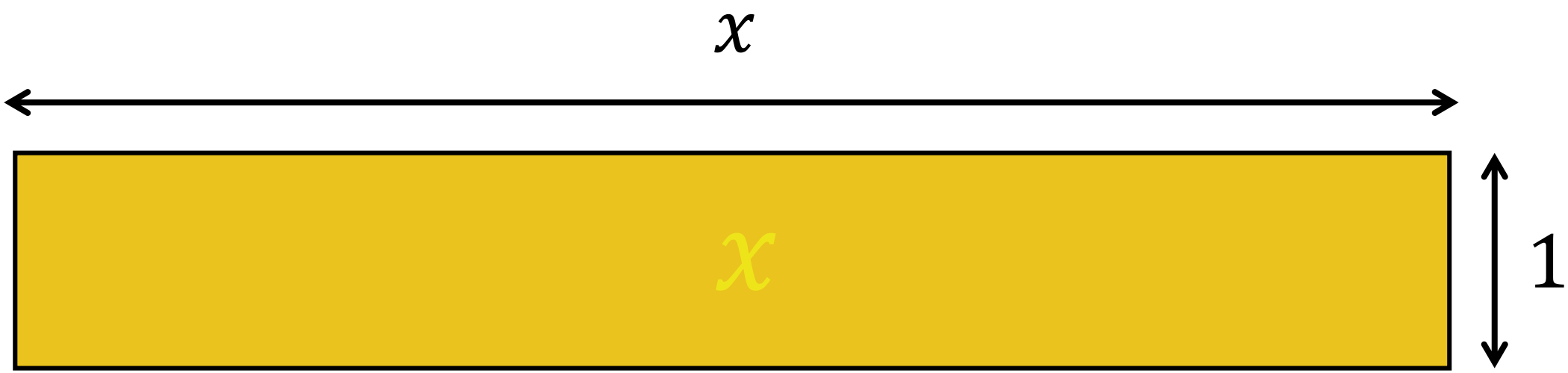
$$\begin{matrix}1\\2\\3\end{matrix}\begin{bmatrix}1 & 1 & 1 & | & 6\\3 & 1 & -1 & | & 8\\9 & 1 & 1 & | & 22\end{bmatrix} \xrightarrow{II:=II-3I} \begin{bmatrix}1 & 1 & 1 & | & 6\\0 & -2 & -4 & | & -10\\9 & 1 & 1 & | & 22\end{bmatrix} \xrightarrow{III:=III-9I} \begin{bmatrix}1 & 1 & 1 & | & 6\\0 & -2 & -4 & | & -10\\0 & -8 & -8 & | & -32\end{bmatrix}$$

$$\xrightarrow{II:=-\frac{II}{2}} \begin{bmatrix}1 & 1 & 1 & | & 6\\0 & 1 & 2 & | & 5\\0 & -8 & -8 & | & -32\end{bmatrix} \xrightarrow{III:=III+8II} \begin{bmatrix}1 & 1 & 1 & | & 6\\0 & 1 & 2 & | & 5\\0 & 0 & 8 & | & 8\end{bmatrix} \xrightarrow{I:=I-II} \begin{bmatrix}1 & 0 & -1 & | & 1\\0 & 1 & 2 & | & 5\\0 & 0 & 8 & | & 8\end{bmatrix}$$

$$\xrightarrow{III:=\frac{III}{8}} \begin{bmatrix}1 & 0 & -1 & | & 1\\0 & 1 & 2 & | & 5\\0 & 0 & 1 & | & 1\end{bmatrix} \xrightarrow{I:=I+2III} \begin{bmatrix}1 & 0 & 0 & | & 2\\0 & 1 & 2 & | & 5\\0 & 0 & 1 & | & 1\end{bmatrix} \xrightarrow{II:=II-2III} \begin{bmatrix}1 & 0 & 0 & | & 2\\0 & 1 & 0 & | & 3\\0 & 0 & 1 & | & 1\end{bmatrix}$$

- Bei dem **babylonischen Wurzelziehen** (Heron-**Verfahren** bzw. Heron-**Algorithmus**) handelt es sich um ein einfaches Verfahren, welches die Quadratwurzel einer positiven Zahl x berechnet: $\mathrm{f}(x) := \mathrm{sqrt}(x) := \sqrt[2]{x}_{+}$
- Das Verfahren lässt sich geometrisch gut veranschaulichen:
  Man stelle sich ein Rechteck vor, welches eine Seitenlänge $a = 1$
  und eine andere Seitenlänge $b = x$ besitzt
- Offensichtlich hat es den Flächeninhalt x

Beispiel: Babylonisches Wurzelziehen - Heron-Algorithmus (Skizze I)

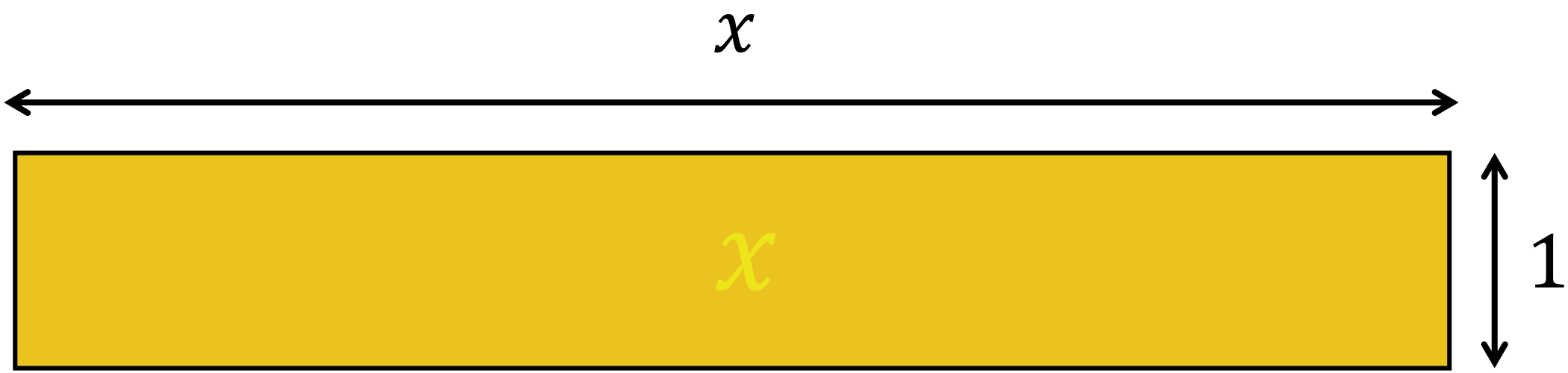


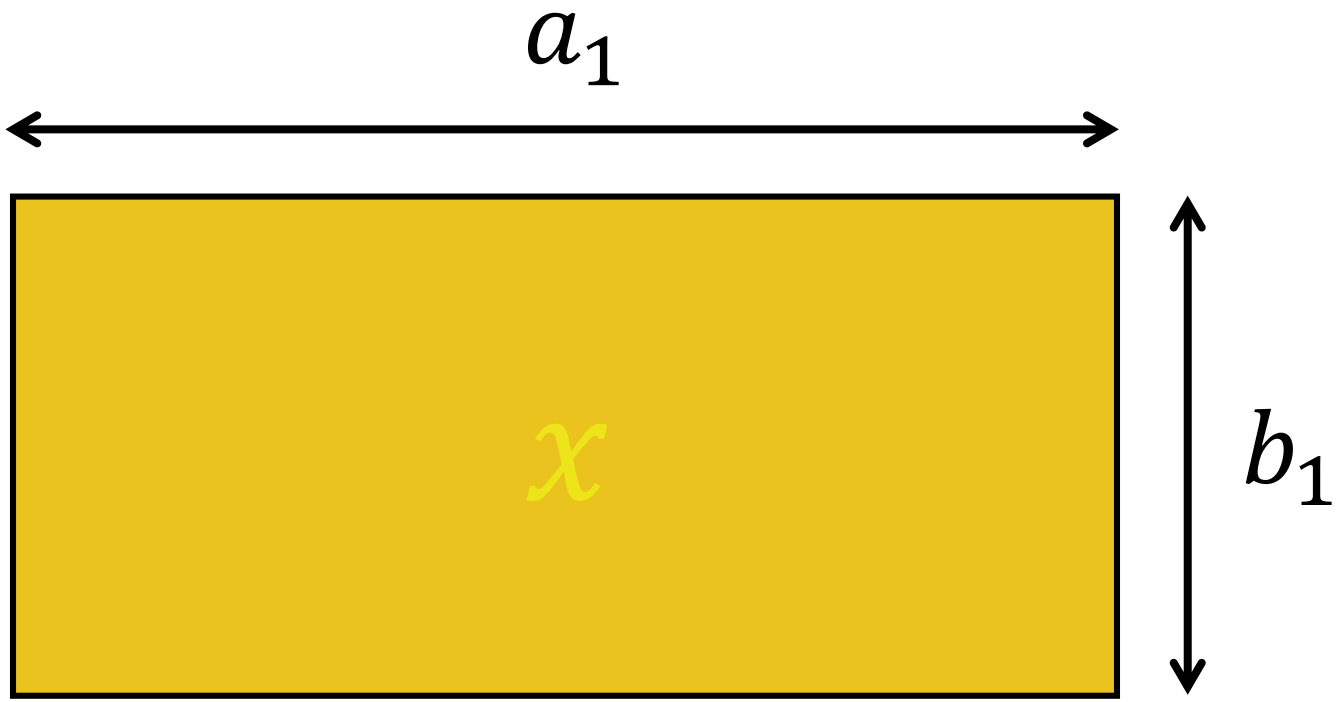
- Nun wollen wir das Rechteck in die Form eines Quadrates mit gleicher Fläche bringen, für dessen Seitenlängen also $a = b = x$ gilt
- Hierzu nehmen wir von beiden Seitenlängen den Mittelwert $\frac{a+b}{2}$ und verwenden diesen als neue Seitenlänge a_1
- Da $a \cdot b = x$ also $b = \frac{x}{a}$ gilt, folgt $a_1 = \frac{a+b}{2} = \frac{a+\frac{x}{a}}{2}$
- Die andere Seitenlänge b_1 muss natürlich so gewählt werden, dass die Fläche gleich bleibt, also weiterhin $a_1 \cdot b_1 = x$ gilt

$$b_1 = \frac{x}{a_1} = \frac{2x^2}{ax + a}$$

- Nun wollen wir das Rechteck in die Form eines Quadrates mit gleicher Fläche bringen, für dessen Seitenlängen also $a = b = x$ gilt
- Hierzu nehmen wir von beiden Seitenlängen den Mittelwert $\frac{a+b}{2}$ und verwenden diesen als neue Seitenlänge a_1
- Da $a \cdot b = x$ also $b = \frac{x}{a}$ gilt, folgt $a_1 = \frac{a+b}{2} = \frac{a+\frac{x}{a}}{2}$
- Die andere Seitenlänge b_1 muss natürlich so gewählt werden, dass die Fläche gleich bleibt, also weiterhin $a_1 \cdot b_1 = x$ gilt
- Das Rechteck mit den neuen Seitenlängen a_1, b_1 ist nun einem Quadrat viel ähnlicher

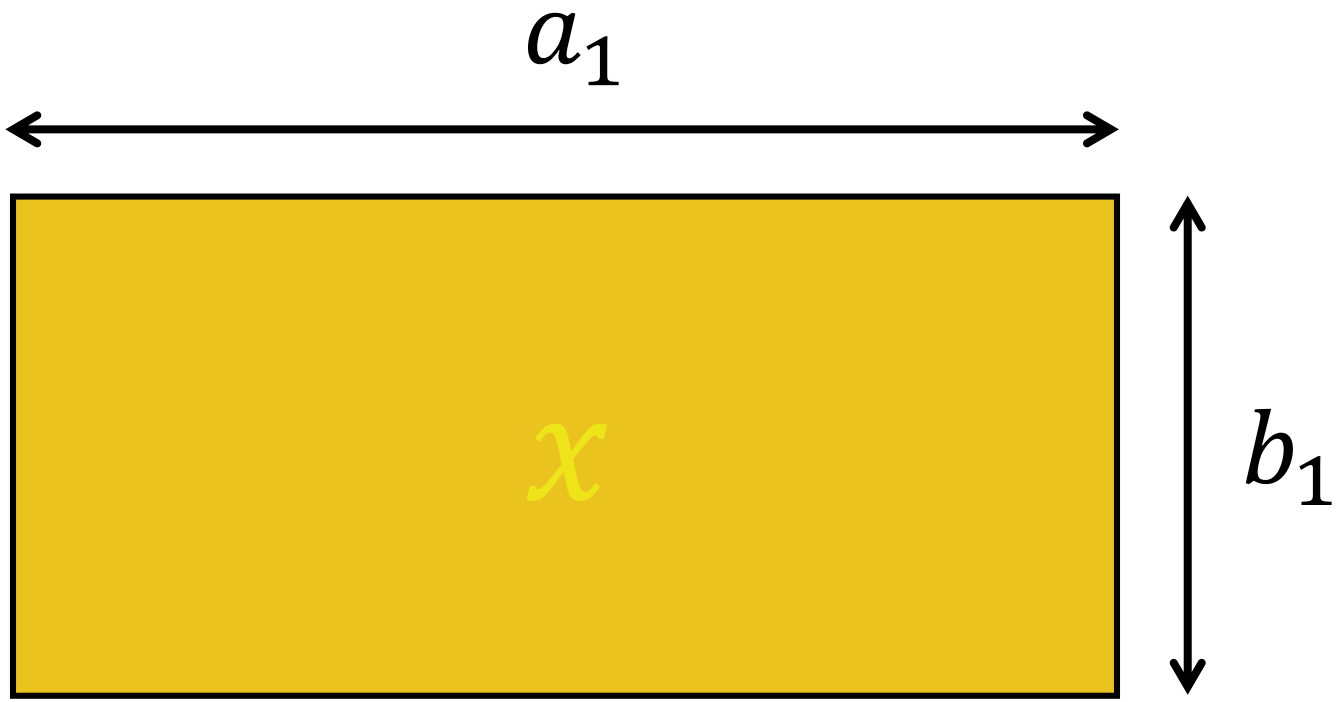
Beispiel: Babylonisches Wurzelziehen - Heron-Algorithmus (Skizze I)

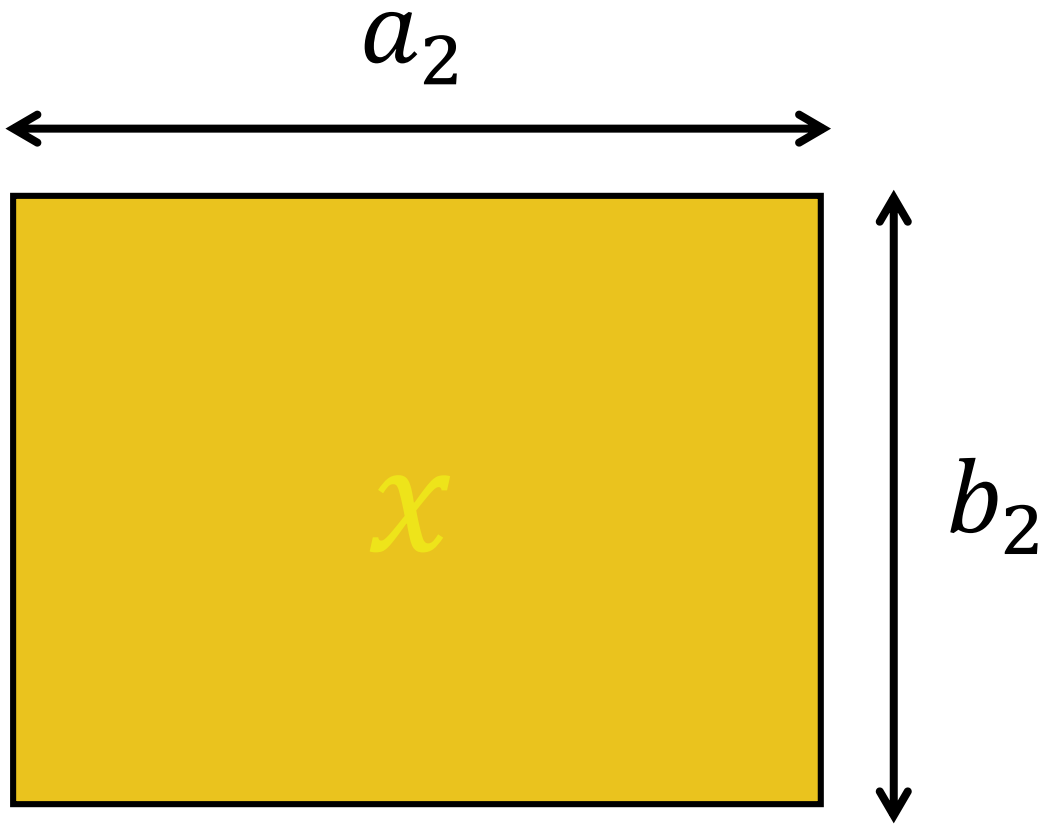


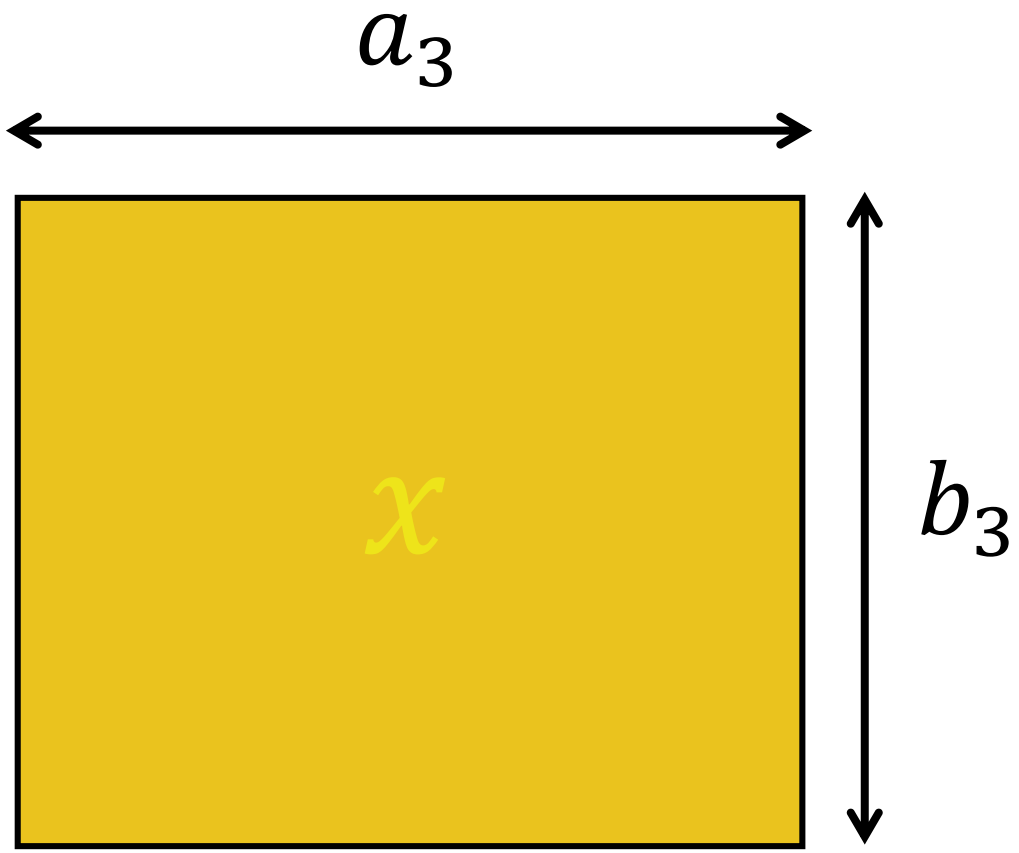

a_1
x
b_1

- Wiederholt man den ganzen Vorgang für die neuen Seitenlängen, ergibt sich ein Rechteck mit Seitenlängen a_2, b_2 das einem Quadrat noch viel ähnlicher ist und man kann sogar zeigen, dass beide Seitenlängen bei unendlicher Wiederholung für alle $x > 0$ gegen x **konvergieren**
- Es reicht für die Berechnung der Quadratwurzel aus, einer der beiden Seitenlängen des Recktecks zu betrachten:

$$a_{i+1} := \frac{a_i + \frac{x}{a_i}}{2} \quad \rightarrow \quad \sqrt[2]{x}$$


a_1
x
b_1


a_2
x
b_2



… und so weiter …

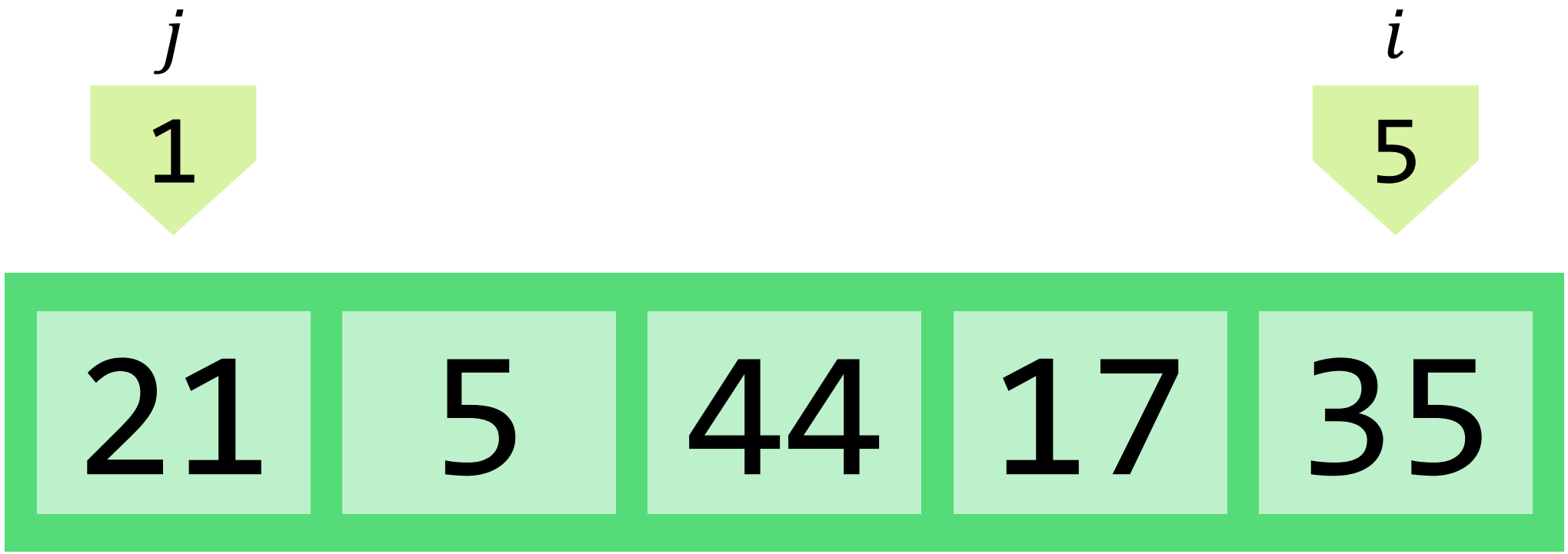
- Die Folge $a_{i+1} := \frac{a_i + \frac{x}{a_i}}{2}$ konvergiert zwar gegen $\sqrt[2]{x}$, damit der rechts stehende Algorithmus aber auch terminiert, wird eine Haltebedingung benötigt
- Der Algorithmus soll bei $n = N$ terminieren: Je größer N ist, desto genauer ist das Ergebnis
- `return` gibt das Ergebnis zurück und den lässt den Algorithmus terminieren

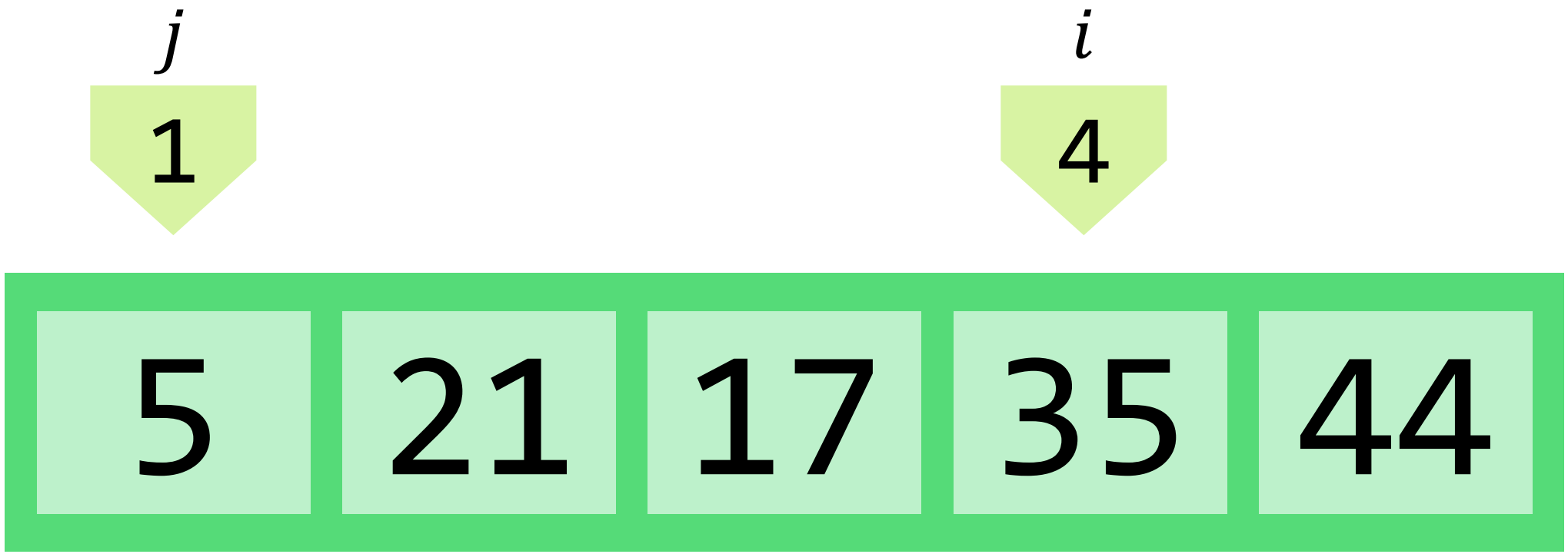
```
sqrt(x)
1  if 0 ≤ x : return 0
2  a := x
3  n := 1
4  while n < N :
5     n := n + 1
6     b := x/a
7     a := a + b
8     a := a/2
9  return a
```

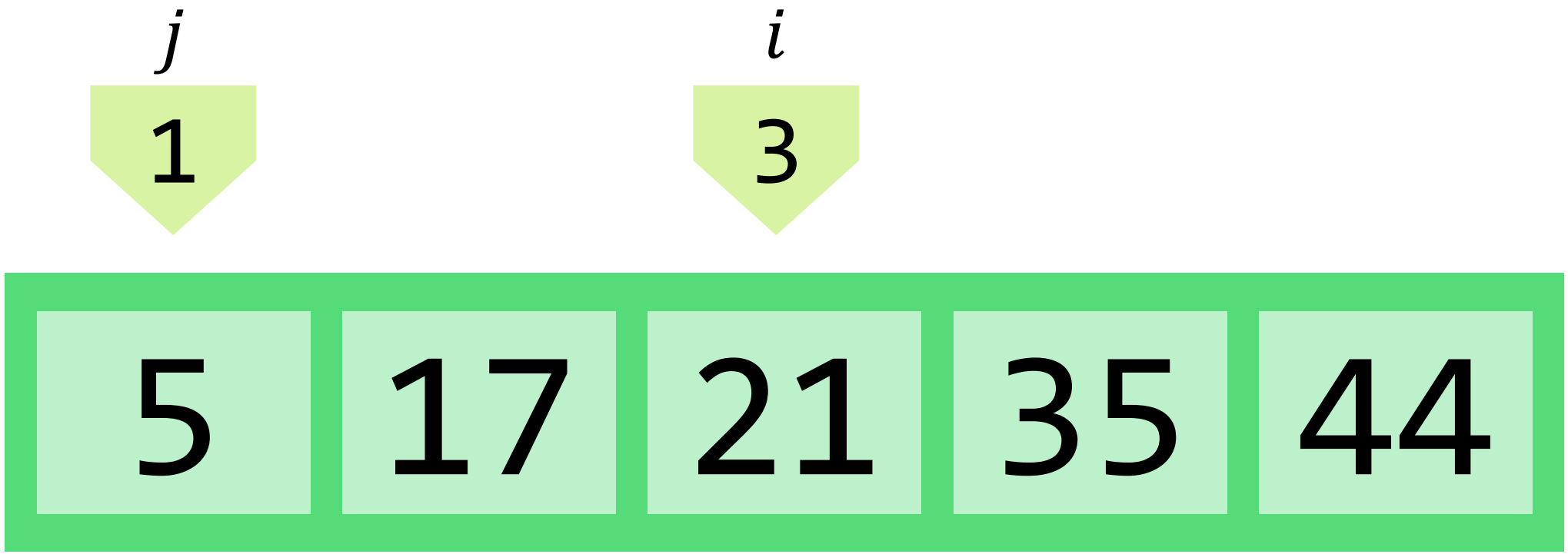
```
bubblesort(a[1], a[2] ··· a[n])
1   i := n
2   while i > 1 :
3       j := 1
4       while i > j :
5           if a[j] > a[j + 1]:
6               swap(a[j], a[j + 1])
7           j := j + 1
8       i := i − 1
```

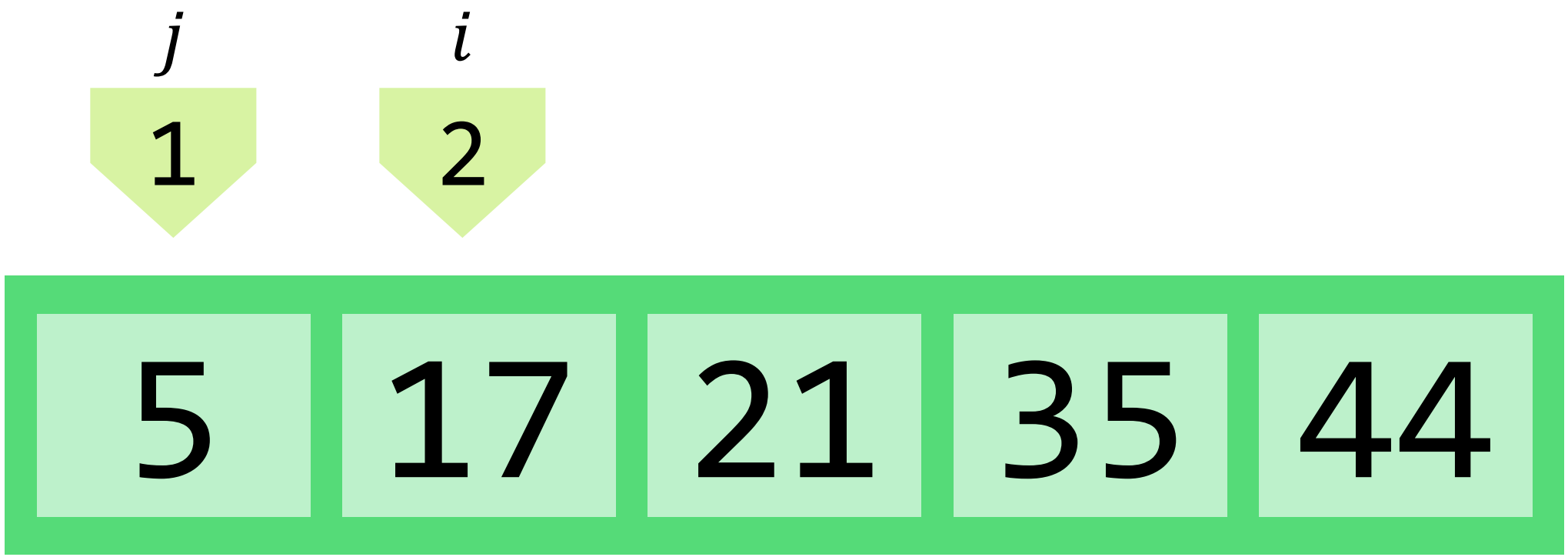
```
swap(a[i], a[j])
1   a[0] := a[i]
2   a[i] := a[j]
3   a[j] := a[0]
```

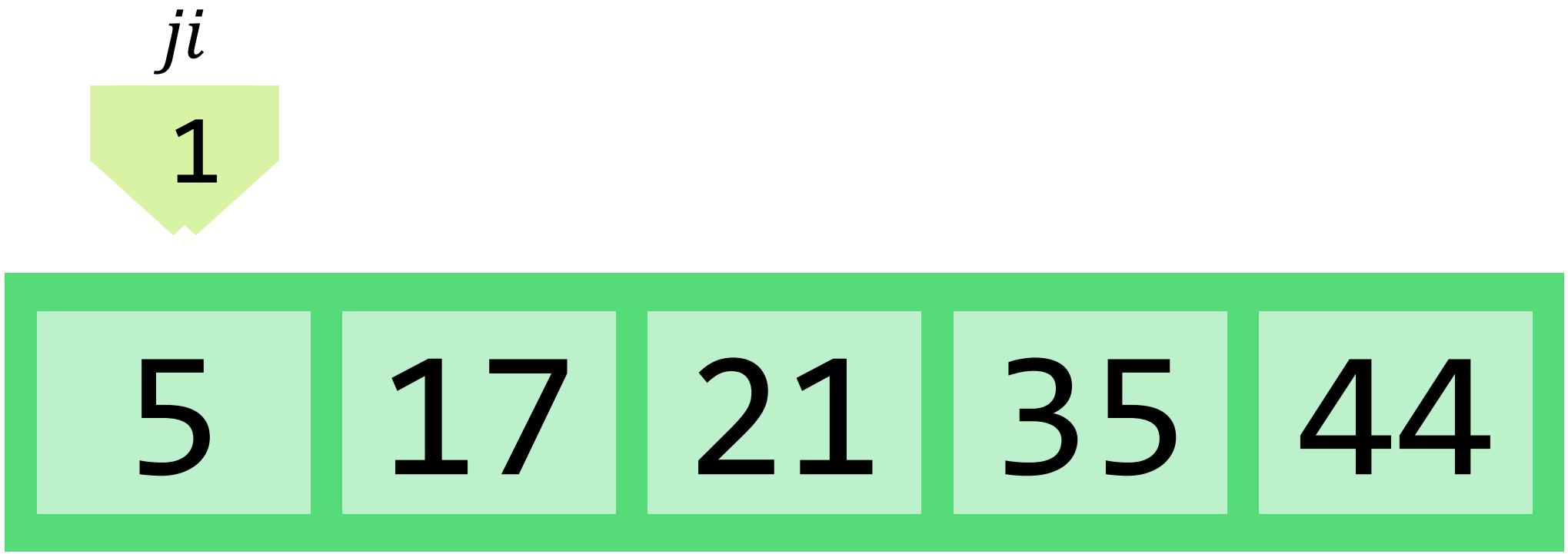
- Bei dem Algorithmus handelt es sich um den sogenannten `bubblesort`-Algorithmus, der eine Folge von Zahlen (Feld) aufsteigend sortiert
- `swap( …, … )` tauscht benachbarte Elemente innerhalb des Feldes
- Große Zahlen steigen bei der Ausführung des Algorithmus im Feld auf, so wie auch Luftblasen im Wasser aufsteigen!
- Der Algorithmus terminiert mit $i = 1$


j
1
i
5
21
5
44
17
35


j
1
i
4
5
21
17
35
44


j
1
i
3
5
17
21
35
44


j
i
1
2
5
17
21
35
44


ji
1
5
17
21
35
44

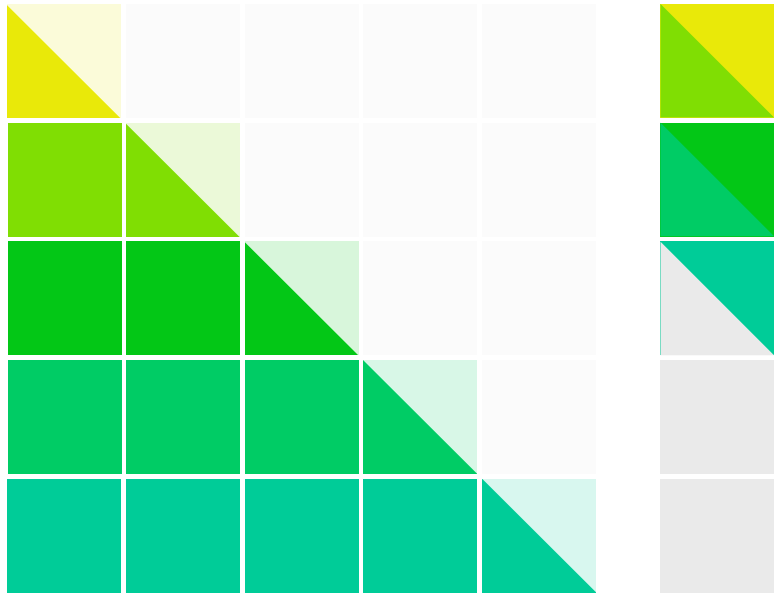
- Es soll gezeigt werden, dass eine Aussage $\mathscr{A}(n)$ für alle $n \in \mathbb{N}$ gilt
- Durch testen einzelner n kann man zeigen, dass eine nicht leere Menge $M \subseteq \mathbb{N}_0$ existiert, so dass $\forall n \in M.\ \mathscr{A}(n)$ gilt: Insbesondere sei $0 \in M$
- Wir wollen nun zeigen, dass $M = \mathbb{N}_0$ gilt!
- Nach den $5.$ Peano-Postulat gilt $M = \mathbb{N}_0$, wenn man zeigen kann, dass wenn $n \in M$ auch für den Nachfolger stets $\boldsymbol{\phi}(n) \in M$ gilt:

$$0 \in M \wedge \left(n \in \mathbb{N} \Rightarrow (n \in M \Rightarrow \boldsymbol{\phi}(n) \in M)\right) \Rightarrow M = \mathbb{N}$$

$$\mathscr{A}(0) \wedge (n.\ \mathscr{A}(n) \Rightarrow \mathscr{A}(\boldsymbol{\phi}(n))) \Rightarrow\ \forall n \in M.\ \mathscr{A}(n)$$

Das Aufsummieren der Zahlen von 1 bis n lässt sich mit einer einfachen Formel berechnen, die als **Gaußsche Summenformel** bekannt ist.

- Ein einfaches Beispiel um eine allgemeine Formel herzuleiten: $\mathrm{sum}(5)$



Gaußsche Summenformel:

$$\mathrm{sum}(n) := 1 + 2 + \cdots + n = \sum_{j=0}^{n} j \overset{!}{=} \frac{n \cdot (n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$$

$$\mathrm{sum}(5) = 1 + 2 + 3 + 4 + 5 = \frac{5 \cdot (5+1)}{2} = \frac{5 \cdot 6}{2} = \frac{30}{2} = 15$$

$$\mathrm{sum}(100) = \frac{100 \cdot (100+1)}{2} = \frac{100 \cdot 101}{2} = \frac{10100}{2} = 5050$$

Induktionsanfang: Wir zeigen, dass die Summenformel für $n = 1$ gilt:

$$\sum_{j=1}^{1} j = 1$$

$$\frac{1 \cdot (1+1)}{2} = \frac{1 \cdot 2}{2} = 1$$

Induktionsvoraussetzung: Wir nehmen an, dass die Summenformel für eine beliebige, aber feste natürliche Zahl n gültig ist.

Induktionsschritt: Wir zeigen, dass die Summenformel auch für den Nachfolger von n, also $n+1$ gilt. Z.z.: $\sum_{j=1}^{n+1} j^{1} = \frac{(n+1)\cdot(n+2)}{2}$:

$$\sum_{j=1}^{n+1} j = (n+1) + \sum_{j=1}^{n} j =_{\text{IV}} (n+1) + \frac{n \cdot (n+1)}{2}$$

$$= \frac{2 \cdot (n+1)}{2} + \frac{n \cdot (n+1)}{2} = \frac{2 \cdot (n+1) + n \cdot (n+1)}{2}$$

$$= \frac{(n+1) \cdot n + (n+1) \cdot 2}{2} = \frac{(n+1) \cdot (n+2)}{2}$$

□

	Anweisung	Kosten	Häufigkeit
1	$i \coloneqq n$	1	1
2	`while` $i > 1$:	1	n
3	$j \coloneqq 1$	1	$n-1$
4	`while` $i > j$:	1	$\sum_{i=2}^{n} i$
5	`if` a$[j] >$ a$[j+1]$:	1	$\sum_{i=2}^{n}[i-1]$
6	`swap`(a$[j]$, a$[j+1]$)	3	0 … $3\sum_{i=1}^{n}[i-1]$
7	$j \coloneqq j+1$	1	$\sum_{i=2}^{n}[i-1]$
8	$i \coloneqq i-1$	1	$n-1$

$$\mathrm{t}_{\blacksquare}(n) \coloneqq \mathrm{t}_{\texttt{bubblesort}}\,(\mathrm{a}[1] \geqslant \mathrm{a}[2] \geqslant \cdots \geqslant \mathrm{a}[n])$$

$$\mathrm{t}_{\mathrm{wc}}(n) = 1 + n + (n-1) + \sum_{i=2}^{n}[i] + 5\sum_{i=1}^{n}[i-1]$$

$$= 1 + n + (n-1) + \sum_{i=0}^{n}[i] - 1 + 5\sum_{i=0}^{n-1}[i]$$

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

$$\mathrm{t}_{\blacksquare}(n) \coloneqq \mathrm{t}_{\texttt{bubblesort}}\,(\mathrm{a}[1] \geqslant \mathrm{a}[2] \geqslant \cdots \geqslant \mathrm{a}[n])$$

$$\mathrm{t}_{\mathrm{bc}}(n) = 1 + n + (n-1) + \sum_{i=2}^{n}[i] + 3\sum_{i=1}^{n}[i-1]$$

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

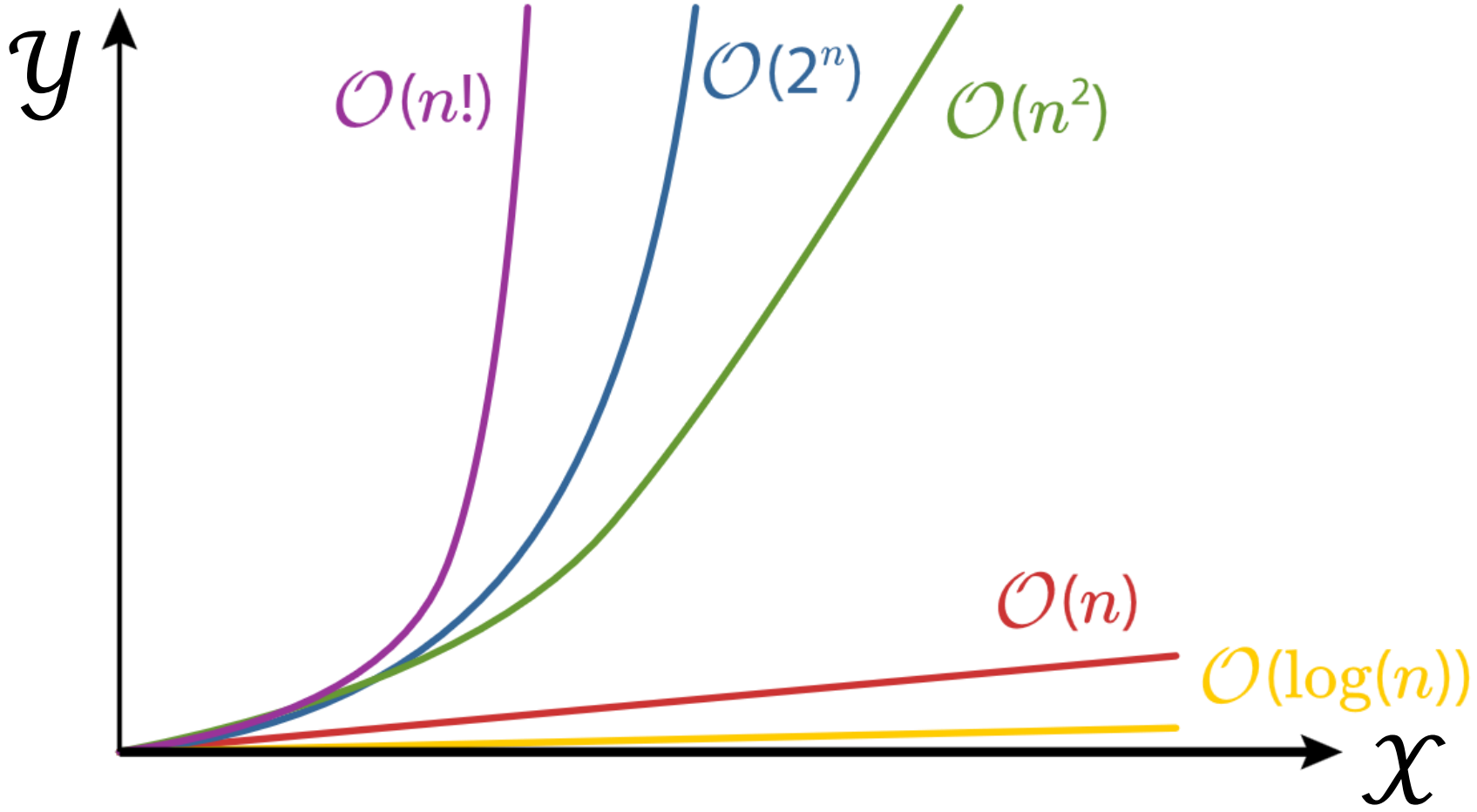
$$= 1 + n + (n-1) + \frac{n \cdot (n+1)}{2} - 1 + 3\frac{(n-1) \cdot n}{2} = \frac{3}{2}n^2 + \frac{5}{2}n - 2$$

- **Obere asymptotische Schranke**, $\mathcal{O}$**-Kalkül** (f wächst höchstens so schnell wie g):

$$\mathrm{f} \in \mathcal{O}(\mathrm{g}) \overset{\mathrm{def}}{\Leftrightarrow} \limsup_{n\to\infty} \frac{\mathrm{f}(n)}{\mathrm{g}(n)} < \infty$$

- **Untere asymptotische Schranke** (f wächst mindestens so schnell wie g):

$$\mathrm{f} \in \Omega(\mathrm{g}) \overset{\mathrm{def}}{\Leftrightarrow} 0 < \liminf_{n\to\infty} \frac{\mathrm{f}(n)}{\mathrm{g}(n)}$$


y
$\mathcal{O}(n!)$
$\mathcal{O}(2^n)$
$\mathcal{O}(n^2)$
$\mathcal{O}(n)$
$\mathcal{O}(\log(n))$
x

- **Exakte asymptotische Schranke** (f wächst genau so schnell wie g):

$$f \in \Theta(g) \overset{\text{def}}{\Leftrightarrow} 0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

- **Wachstum gleicher Ordnung** (f wächst zur gleichen Ordnung wie g):

$$f \sim g \overset{\text{def}}{\Leftrightarrow} \lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$$

- **Starke obere Schranke** (f wächst langsamer als g):

$$f \in o(g) \overset{\text{def}}{\Leftrightarrow} \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Leftrightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = \infty$$

- **Starke untere asymptotische Schranke** (f wächst schneller als g):

$$f \in \omega(g) \overset{\text{def}}{\Leftrightarrow} \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Leftrightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$

$\lim_{n\to\infty} \frac{f(n)}{g(n)}$ Grenzwert des Quotienten	$f \in o(g)$ f wächst langsamer als g $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	$f \in \mathcal{O}(g)$ f wächst höchstens so schnell wie g $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$	$f \in \Theta(g)$ f wächst genau so schnell wie g $0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$	$f \in \Omega(g)$ f wächst mindestens so schnell wie g $0 < \lim_{n\to\infty} \frac{f(n)}{g(n)}$	$f \in \omega(g)$ f wächst schneller als g $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	✓	✓			
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 42$		✓	✓	✓	
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$				✓	✓
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \perp$					

$$\lim_{n\to\infty} \mathrm{f}(n), \mathrm{g}(n) = \infty, 0 \Rightarrow \lim_{n\to\infty} \frac{\mathrm{f}(n)}{\mathrm{g}(n)} = \lim_{n\to\infty} \frac{\frac{\mathrm{d}}{\mathrm{d}n}[\mathrm{f}(n)]}{\frac{\mathrm{d}}{\mathrm{d}n}[\mathrm{g}(n)]}$$

$$t_{\blacksquare} \in \Theta(n^2)$$

$$0 < \lim_{n\to\infty} \frac{t_{wc}(n)}{n^2} = \lim_{n\to\infty} \frac{3n^2 + n - 2}{n^2} = \lim_{n\to\infty} \frac{6n + 1}{2n} = \frac{6}{2} = 3 < \infty$$

$$0 < \lim_{n\to\infty} \frac{t_{bc}(n)}{n^2} = \lim_{n\to\infty} \frac{\frac{3}{2}n^2 + \frac{5}{2}n - 2}{n^2} = \lim_{n\to\infty} \frac{3n + 2.5}{2n} = \frac{3}{2} = 1.5 < \infty$$