

# Lösung zum Präsenzblatt

zur Vorlesung

## Grundzüge der Informatik I

Besprechung in Kalenderwoche 15.

### Aussagenlogik – Grundlagen

**Aussagen:** z. B.

$A$  = die Eingabe ist groß

$B$  = Der Algorithmus löst das Problem

mit Wahrheitsbelegungen 0 oder 1

**Verknüpfungen:**

unär: z.,B. Negation  $\neg$

binär: z.,B. Konjunktion  $\wedge$ , Disjunktion  $\vee$ , Implikation  $\Rightarrow$ , Äquivalenz  $\Leftrightarrow$

$A$	$B$	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

**Aufgabe 1** Aussagenlogik

$O$	$K$	9	6
?	egal	egal	?

Vokal	ungerade	$Vokal \Rightarrow ungerade$
0	0	1
0	1	1
1	0	0
1	1	1

$A$  = die Eingabe ist groß

$B$  = Der Algorithmus löst das Problem

$A \Rightarrow B$

$A$	$B$	$A \Rightarrow B$	$A \wedge B$	$B \Rightarrow A$	$\neg A \vee B$	$\neg B \Rightarrow \neg A$
0	0	1	0	1	1	1
0	1	1	0	0	1	1
1	0	0	0	1	0	0
1	1	1	1	1	1	1
			×	×	✓	✓

## Aufgabe 2 Summen-Notation

Gegeben:

- Array  $A = [a_1, a_2, \dots, a_n]$  aufsteigend sortiert  
( $a_i \in \mathbb{N}$ ,  $1 \leq i \leq n$ , und  $a_1 \leq a_2 \leq \dots \leq a_n$ ),
- Index  $k$ ,  $1 \leq k \leq \lfloor n/2 \rfloor$ .

a) Zu zeigen:

$$\sum_{i=1}^k a_i \geq 10 \implies \sum_{i=k+1}^n a_i \geq 10.$$

b) Zusätzlich:  $a_i$  unterschiedlich ( $a_i \neq a_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq n$ ).

Zu zeigen:

$$\sum_{i=k+1}^n a_i \geq k^2.$$

a)

Array  $A = [a_1, a_2, \dots, a_n]$ ,  $a_1 \leq a_2 \leq \dots \leq a_n$ , Index  $k$ ,  $1 \leq k \leq \lfloor n/2 \rfloor$ .

Zu zeigen:

$$\sum_{i=1}^k a_i \geq 10 \implies \sum_{i=k+1}^n a_i \geq 10.$$

Es gelten  $a_i \leq a_j$  für  $i < j$  und  $k \leq \lfloor n/2 \rfloor \implies$

Für jedes  $a_i$ ,  $1 \leq i \leq k$ , existiert  $a_{k+i}$  im Array und  $a_i \leq a_{k+i}$ .

Damit gilt:

$$\begin{aligned} \sum_{i=1}^k a_i &= a_1 + a_2 + \dots + a_k \\ &\leq a_{k+1} + a_{k+2} + \dots + a_{2k} = \sum_{i=k+1}^{2k} a_i \leq \sum_{i=k+1}^n a_i. \end{aligned}$$

Daraus folgt:

$$\sum_{i=1}^k a_i \geq 10 \implies \sum_{i=k+1}^n a_i \geq 10. \square$$

b)

Array  $A = [a_1, a_2, \dots, a_n]$ ,  $a_1 < a_2 < \dots < a_n$ , Index  $k$ ,  $1 \leq k \leq \lfloor n/2 \rfloor$ .

Zu zeigen:

$$\sum_{i=k+1}^n a_i \geq k^2.$$

Da  $a_1 < a_2 < \dots < a_{k+1}$  und  $a_i \in \mathbb{N}$ , gilt  $a_{k+1} \geq k$ .

Da  $n \geq 2k$ , kommen in  $\sum_{i=k+1}^n$  mindestens  $k$  Summanden  $a_j \geq a_{k+1} \geq k$  vor.

Also gilt insgesamt

$$\sum_{i=k+1}^n a_i \geq k \cdot k = k^2. \square$$

### Diskussion

- Nachvollziehen der logischen Struktur der Aufgabenstellung und der Beweisskizze.
- Äquivalent zur Aussage

$$\sum_{i=1}^k a_i \geq 10 \implies \sum_{i=k+1}^n a_i \geq 10$$

kann auch

$$\sum_{i=k+1}^n a_i < 10 \implies \sum_{i=1}^k a_i < 10$$

gezeigt werden.

- Es gilt **nicht**  $\sum_{i=k+1}^n a_i \geq 10 \implies \sum_{i=1}^k a_i \geq 10$

# Pseudocode

Die Vorlesung verwendet Pseudocode, um Algorithmen formal, aber unabhängig von spezifischen Programmiersprachen darzustellen. Die Verwendete Syntax ist in der Vorlesung definiert und soll hier geübt werden.

Tipps, auf was Sie beim Schreiben von Pseudocode achten sollten, um häufige Fehler zu vermeiden:

- Benennen Sie Ihre Methode und entnehmen Sie der Aufgabe, welche Variablen gegeben sind. Die Länge von Feldern, kann immer als Parameter mit übergeben werden.
- Rücken Sie Codeblöcke ein, um eindeutig zu repräsentieren, wann Bedingungen und Schleifen gelten.
- Verwenden Sie immer den **return** Befehl, um das Ergebnis Ihres Algorithmus auszugeben. Achten Sie dabei darauf, ob das Ergebniss die von der Aufgabe geforderte Form hat.
- Achten Sie darauf, dass alle Ihre Variablen entweder Parameter ihres Algorithmus sind oder initialisiert werden. Eine Variable kann z.B. nicht mit etwas verglichen werden, wenn ihr nicht vorher ein Wert zugewiesen wurde.
- Sie können alle mathematischen Ausdrücke in Kombination mit Ihren Variablen verwenden, wie z.B.  $a^i$ ,  $\log_3 a$ ,  $a!$  oder  $\max\{a, 3\}$ , wobei  $a$  und  $i$  zuvor definierte Variablen sind. Wir raten allerdings davon ab Notationen zu verwenden, die Schleifen beinhalten, da diese schnell zu Fehlern in Laufzeitanalysen führen können. Dies sind zum Beispiel die Summennotation  $\sum$ , die Produktnotation  $\prod$  oder große Mengenoperationen wie  $\cup$ .

## Aufgabe 3 Pseudocode

- a) Gegeben ein Feld  $A$  der Länge  $n$ . Schreiben Sie einen Algorithmus in Pseudocode, der die Summe aller Elemente aus  $A$  ausgibt.

Summe( $A, n$ ):

1.  $s = 0$
2. **for**  $i = 1$  **to**  $n$  **do**
3.      $s = s + A[i]$
4. **return**  $s$

- b) Gegeben ein Feld  $A$  der Länge  $n$ . Verwenden Sie Ihren Algorithmus aus der vorherigen Teilaufgabe, um den Durchschnitt der Elemente aus  $A$  zu berechnen.

Durchschnitt( $A, n$ ):

1.  $d = \text{Summe}(A, n)$
2.  $d = d/n$
3. **return**  $d$

- c) Gegeben ein Feld  $A$  der Länge  $n$  mit den Werten  $a_1, a_2, \dots, a_n$  und eine Zahl  $m \in \mathbb{N}$ . Sei  $k$  die größte Zahl, so dass gilt  $\sum_{i=1}^k a_i \leq m$ . Entwickeln Sie einen Algorithmus, der  $k$  berechnet.

Packing( $A, n, m$ ):

1.  $d = m$
2.  $i = 1$
3. **while**  $d \geq A[i]$  **do**
4.      $d = d - A[i]$
5.      $i = i + 1$
6. **return**  $i - 1$