

Algorithmen und Programmierung

13. Übung – Lösungsvorschläge

1. Aufgabe:

Das Ausgangsfeld enthält nur ganze Zahlen aus der Menge $\{0, \dots, n-1\}$. Wir erzeugen uns daher ein Hilfsfeld, mit dem wir zählen, wie oft jede Zahl im Ausgangsfeld vorkommt. Der Index des Hilfsfeldes entspricht dabei der Zahl, deren Häufigkeit gezählt wird, und der Inhalt an dieser Stelle der Anzahl im Ausgangsfeld.

```
public static void doppelteAusgeben(int [] feld) {  
    int [] anz;  
    int i;  
  
    // Hilfsfeld fuer Anzahlen  
    anz = new int[feld.length];  
    for (i=0; i<anz.length; i++)  
        anz[i]=0;  
  
    // Zaehlen  
    for (i=0; i<feld.length; i++)  
        anz[feld[i]]++;  
  
    // Ausgeben  
    for (i=0; i<anz.length; i++)  
        if (anz[i] == 2)  
            System.out.println(i + "_ist_2mal_enthalten");  
    return;  
}
```

Unsere Methode enthält drei (nicht verschachtelte) FOR-Schleifen (die erste zur Initialisierung könnte man in Java weglassen), die jeweils n Durchläufe haben. Der Aufwand beträgt folglich

$$O(3 \cdot n) = O(n).$$

2. Aufgabe:

- a) Nach Abarbeitung des Programmstücks enthält die Variable y das Quadrat des für x eingelesenen Wertes.
- b) Seien x_{start} der eingelesene Wert für x sowie x_l , y_l und z_l die Werte der Variablen x , y und z nach dem l -ten Schleifendurchlauf. Dabei ist $z_l = x_{start} \forall l$, da sich der Wert der Variablen z während der Abarbeitung der Schleife nicht verändert.

- 1) Endlichkeit der Anzahl der Schleifendurchläufe n

x_{start} ist eine endliche, ganze Zahl ≥ 0 . Die Schleife bricht ab, wenn der Wert von x gleich 0 ist. In jedem Schleifendurchlauf wird der Wert von x um 1 verringert. Folglich ist die Anzahl der Schleifendurchläufe endlich mit $n = x_{start}$.

- 2) Beweis, dass $x_{start} \cdot x_{start}$ berechnet wird

Invariante:

$$x_{start} \cdot x_{start} = y_l + x_l \cdot x_{start} \quad (\text{bzw.} \quad y_l = l \cdot x_{start})$$

Induktionsanfang:

Für $l = 0$ gilt: $x_0 = x_{start}$ und $y_0 = 0$

Überprüfung der Invariante:

$$\begin{aligned} y_0 + x_0 \cdot x_0 &= 0 + x_{start} \cdot x_{start} \\ &= x_{start} \cdot x_{start} \end{aligned}$$

Die Invariante gilt.

Induktionsschritt:

Sei $1 \leq l \leq n$ fest. Dann gilt für $l - 1$ nach Induktionsvoraussetzung (Invariante):

$$x_{start} \cdot x_{start} = y_{l-1} + x_{l-1} \cdot x_{start}$$

Wir betrachten nun den l -ten Schleifendurchlauf:

$$\begin{aligned} x_l &= x_{l-1} - 1 \\ y_l &= y_{l-1} + z_l \\ &= y_{l-1} + x_{start} \end{aligned}$$

Damit erhalten wir

$$\begin{aligned} y_l + x_l \cdot x_{start} &= y_{l-1} + x_{start} + (x_{l-1} - 1) \cdot x_{start} \\ &= y_{l-1} + x_{start} + x_{l-1} \cdot x_{start} - x_{start} \\ &= y_{l-1} + x_{l-1} \cdot x_{start} \\ &= x_{start} \cdot x_{start} \end{aligned}$$

und die Invariante gilt.

3) Quintessenz

Nach dem letzten (n -ten) Schleifendurchlauf ist die Schleifenbedingung nicht mehr erfüllt ($x_n = 0$) und die Invariante gilt, also

$$x_{start} \cdot x_{start} = y_n + x_n = y_n$$

Folglich ist in der Variable y nach Abarbeitung des Programmstücks der Wert $x_{start} \cdot x_{start}$ gespeichert. Die Aussage von a) ist also korrekt.

3. Aufgabe:

- a) Nach Abarbeitung des Programmstücks enthält die Variable y die Summe aus dem eingelesenen Wert für y und dem Doppelten des eingelesenen Wertes für x .
- b) Seien x_{start} und y_{start} die eingelesenen Werte für x und y sowie x_l und y_l die Werte der Variablen x und y nach dem l -ten Schleifendurchlauf.

- 1) Endlichkeit der Anzahl der Schleifendurchläufe n

x_{start} ist eine endliche, ganze Zahl ≥ 0 . Die Schleife bricht ab, wenn der Wert von x gleich 0 ist. In jedem Schleifendurchlauf wird der Wert von x um 1 verringert. Folglich ist die Anzahl der Schleifendurchläufe endlich mit $n = x_{start}$.

- 2) Beweis, dass $y_{start} + 2 \cdot x_{start}$ berechnet wird

Invariante:

$$y_{start} + 2 \cdot x_{start} = y_l + 2 \cdot x_l$$

Induktionsanfang:

Für $l = 0$ gilt: $x_0 = x_{start}$ und $y_0 = y_{start}$

Überprüfung der Invariante:

$$y_0 + 2 \cdot x_0 = y_{start} + 2 \cdot x_{start}$$

Die Invariante gilt.

Induktionsschritt:

Sei $1 \leq l \leq n$ fest. Dann gilt für $l - 1$ nach Induktionsvoraussetzung (Invariante):

$$y_{start} + 2 \cdot x_{start} = y_{l-1} + 2 \cdot x_{l-1}$$

Wir betrachten nun den l -ten Schleifendurchlauf:

$$\begin{aligned} x_l &= x_{l-1} - 1 \\ y_l &= y_{l-1} + 2 \end{aligned}$$

Damit erhalten wir

$$\begin{aligned}y_l + 2 \cdot x_l &= y_{l-1} + 2 + 2 \cdot (x_{l-1} - 1) \\ &= y_{l-1} + 2 + 2 \cdot x_{l-1} - 2 \\ &= y_{l-1} + 2 \cdot x_{l-1} \\ &= y_{start} + 2 \cdot x_{start}\end{aligned}$$

und die Invariante gilt.

3) Quintessenz

Nach dem letzten (n -ten) Schleifendurchlauf ist die Schleifenbedingung nicht mehr erfüllt ($x_n = 0$) und die Invariante gilt, also

$$y_{start} + 2 \cdot x_{start} = y_n + 2 \cdot x_n = y_n$$

Folglich ist in der Variable y nach Abarbeitung des Programmstücks der Wert $y_{start} + 2 \cdot x_{start}$ gespeichert. Die Aussage von a) ist also korrekt.

4. Aufgabe:

- Nach Abarbeitung des Programmstücks enthält die Variable y die Summe aller Zahlen von 1 bis zum eingelesenen Wert für x .
- Seien x_{start} der eingelesene Wert für x sowie x_l und y_l die Werte der Variablen x und y nach dem l -ten Schleifendurchlauf.

1) Endlichkeit der Anzahl der Schleifendurchläufe n

x_{start} ist eine endliche, ganze Zahl ≥ 0 . Die Schleife bricht ab, wenn der Wert von x gleich 0 ist. In jedem Schleifendurchlauf wird der Wert von x um 1 verringert. Folglich ist die Anzahl der Schleifendurchläufe endlich mit $n = x_{start}$.

- Beweis, dass $\sum_{i=1}^{x_{start}} i = \frac{x_{start} \cdot (x_{start} + 1)}{2}$ berechnet wird

Invariante:

$$\frac{x_{start} \cdot (x_{start} + 1)}{2} = y_l + \frac{x_l \cdot (x_l + 1)}{2}$$

Induktionsanfang:

Für $l = 0$ gilt: $x_0 = x_{start}$ und $y_0 = 0$

Überprüfung der Invariante:

$$y_0 + \frac{x_0 \cdot (x_0 + 1)}{2} = \frac{x_{start} \cdot (x_{start} + 1)}{2}$$

Die Invariante gilt.

Induktionsschritt:

Sei $1 \leq l \leq n$ fest. Dann gilt für $l - 1$ nach Induktionsvoraussetzung (Invariante):

$$\frac{x_{start} \cdot (x_{start} + 1)}{2} = y_{l-1} + \frac{x_{l-1} \cdot (x_{l-1} + 1)}{2}$$

Wir betrachten nun den l -ten Schleifendurchlauf:

$$\begin{aligned} x_l &= x_{l-1} - 1 \\ y_l &= y_{l-1} + x_{l-1} \end{aligned}$$

Damit erhalten wir

$$\begin{aligned} y_l + \frac{x_l \cdot (x_l + 1)}{2} &= y_{l-1} + x_{l-1} + \frac{(x_{l-1} - 1) \cdot (x_{l-1} - 1 + 1)}{2} \\ &= y_{l-1} + \frac{2 \cdot x_{l-1} + x_{l-1}^2 - x_{l-1}}{2} \\ &= y_{l-1} + \frac{x_{l-1}^2 + x_{l-1}}{2} \\ &= y_{l-1} + \frac{x_{l-1} \cdot (x_{l-1} + 1)}{2} \\ &= \frac{x_{start} \cdot (x_{start} + 1)}{2} \end{aligned}$$

und die Invariante gilt.

3) Quintessenz

Nach dem letzten (n -ten) Schleifendurchlauf ist die Schleifenbedingung nicht mehr erfüllt ($x_n = 0$) und die Invariante gilt, also

$$\frac{x_{start} \cdot (x_{start} + 1)}{2} = y_n + \frac{x_n \cdot (x_n + 1)}{2} = y_n$$

Folglich ist in der Variable y nach Abarbeitung des Programmstücks der Wert $\frac{x_{start} \cdot (x_{start} + 1)}{2}$ gespeichert. Die Aussage von a) ist also korrekt.

5. Aufgabe:

a) $-0,25_{10}$

b) $13,75_{10}$

c) -34_{10}

6. Aufgabe:

a)

$$\begin{aligned} 1,0\overline{1}_2 &= 2^0 + 2^{-2} + 2^{-4} + 2^{-6} + \dots \\ &= (2^{-2})^0 + (2^{-2})^1 + (2^{-2})^2 + (2^{-2})^3 + \dots \\ &= \lim_{n \rightarrow \infty} \sum_{i=0}^n (2^{-2})^i \\ &= \frac{1}{1-\frac{1}{4}} \\ &= \frac{4}{3} \end{aligned}$$

b)

$$\begin{aligned} 1,0\overline{01}_2 &= 2^0 + 2^{-3} + 2^{-6} + 2^{-9} + \dots \\ &= (2^{-3})^0 + (2^{-3})^1 + (2^{-3})^2 + (2^{-3})^3 + \dots \\ &= \lim_{n \rightarrow \infty} \sum_{i=0}^n (2^{-3})^i \\ &= \frac{1}{1-\frac{1}{8}} \\ &= \frac{8}{7} \end{aligned}$$

7. Aufgabe:

Zunächst führt das Programm (siehe Lösungsunterlagen 6. Übung) einige Anweisungen mit konstantem Aufwand ($O(1)$) aus. Diese können in der Aufwandsberechnung vernachlässigt werden. Dann wird mit Hilfe einer `while`-Schleife der Wert $\log_2 n$ bestimmt. Diese Bestimmung erfolgt durch systematisches Dividieren und hat folglich einen Aufwand von $O(\log n)$.

Anschließend folgt die Überprüfung, ob n die Form a^b hat, mit drei geschachtelten `while`-Schleifen.

Die äußerste Schleife testet jedes mögliche b (Exponent), wobei b Werte von 2 bis $\log_2 n$ annehmen kann. Die Anzahl der Durchläufe der äußeren Schleife ist folglich logarithmisch in der Eingabegröße n , also $O(\log n)$.

Die nächste (innere) Schleife führt eine binäre Suche für das feste b nach dem möglichen a (Basis) durch. Die binäre Suche teilt das Suchintervall immer in zwei Hälften und sucht in der passenden Hälfte weiter. Dies entspricht dem systematischen Dividieren der Anzahl der Elemente des Startintervalls durch 2, bis man ein 1-elementiges Intervall erreicht hat. Da die Länge des Intervalls von n abhängt, ist der Aufwand dieser Schleife ebenfalls logarithmisch in der Eingabegröße n , also $O(\log n)$.

Innerhalb dieser binären Suche muss nun noch die Potenz der aktuellen Werte der beiden Zählvariablen berechnet werden. Dabei werden im schlimmsten Fall (im letzten Durchlauf der äußeren Schleife hat b den Wert $\log_2 n$) $\log_2 n$ Multiplikationen durchgeführt. Der Aufwand der Potenzberechnung ist also auch $O(\log n)$.

Nachdem die binäre Suche nach a für ein bestimmtes b abgeschlossen ist, wird innerhalb dieses Durchlaufs der äußersten Schleife noch einmal die Potenz des gefundenen a hoch b berechnet. Da b maximal gleich $\log_2 n$ ist, ist auch dieser Aufwand $O(\log n)$.

Der Gesamtaufwand des Algorithmus beläuft sich folglich auf

$$O(\log n + (\log n \cdot ((\log n)^2 + \log n))) = O((\log n)^3).$$