

# **Datenstrukturen und Algorithmen: Übungsblatt #4**

Abgabe am 9. Mai 2018

**Finn Hess (378104), Jan Knichel (377779), Paul Orschau (381085)**

4.5.2018

## Aufgabe 1

Durch Betrachtung des Algorithmus ist klar, dass  $S(1) = 1$  und  $S(2) = 1$  gelten müssen, denn für kleine  $n$  endet der Algorithmus bereits im ersten Schleifendurchlauf in Zeile 7.

Nun betrachten wir die Größe des undurchsuchten Arrays nach einem Schleifendurchlauf. Es gelten folgende Beziehungen (siehe Code):

$$\text{lmid} = \lceil \frac{2l+r}{3} \rceil$$

$$\text{rmid} = \lfloor \frac{l+2r}{3} \rfloor$$

Da das Array gedrittelt wird, ist die Größe des Arrays im nächsten Durchlauf entweder

1.  $\text{lmid} - l = \lceil \frac{2l+r}{3} \rceil - l = \lceil \frac{r-l}{3} \rceil = \lceil \frac{n-1}{3} \rceil \geq \frac{n-1}{3}$
2.  $\text{rmid} - \text{lmid} = \lfloor \frac{l+2r}{3} \rfloor - \lceil \frac{2l+r}{3} \rceil = \lfloor \frac{2n-2}{3} \rfloor - \lceil \frac{n-1}{3} \rceil \leq \frac{2n-2}{3} - \frac{n-1}{3} = \frac{n-1}{3}$
3.  $r - \text{rmid} = r - \lfloor \frac{l+2r}{3} \rfloor = -(-r + \lfloor \frac{l+2r}{3} \rfloor) = -\lfloor \frac{l-r}{3} \rfloor = -\lfloor \frac{1-n}{3} \rfloor = \lceil \frac{n-1}{3} \rceil \geq \frac{n-1}{3}$

Wie man anhand der Abschätzungen sehen kann, sind Fall 1 und 3 stets gleich groß, während Fall 2 stets kleiner bzw. gleich groß ist. Daher müssen wir für die maximale Anzahl von Schleifendurchläufen nur Fall 1 betrachten!

Die Rekursionsgleichung lautet also

$$S(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ 1 & n = 2 \\ 1 + S(\lceil \frac{n-1}{3} \rceil) & n > 2 \end{cases} \quad (1)$$

Betrachten wir nun den Spezialfall  $n = \frac{3^k-1}{2}$  mit  $k > 1$ . Dann gilt

$$S(\frac{3^k-1}{2}) = 1 + S(\lceil \frac{\frac{3^k-1}{2}-1}{3} \rceil) = 1 + S(\lceil \frac{3^k-3}{6} \rceil) = 1 + S(\lceil \frac{3^{k-1}-1}{2} \rceil) = \dots = k + S(\lceil \frac{3^0-1}{2} \rceil) = k \quad (2)$$

Wir vermuten also dass  $S(n) = k$  gilt, falls gilt:

$$\begin{aligned} \frac{3^{k-1}}{2} &\leq n < \frac{3^k}{2} \\ \Leftrightarrow 3^{k-1} &\leq 2n < 3^k \\ \Leftrightarrow k-1 &\leq \log_3(2n) < k \end{aligned} \quad (3)$$

Damit erhalten wir als Lösung für  $S(n) = \lfloor \log_3(2n) \rfloor + 1$ .

Beweis per Induktion.

Induktionsanfang:  $S(1) = 1 = \lfloor \log_3(2 \cdot 1) \rfloor + 1 \checkmark$

Induktionsvoraussetzung: Die obige Gleichung stimmt für ein festes, aber beliebiges  $n \in \mathbb{N}$  mit  $n > 1$ .

Induktionsschritt:

$$\begin{aligned} S(n) &= 1 + S(\lceil \frac{n-1}{3} \rceil) \\ &= 1 + \lfloor \log_3(2 \cdot \lceil \frac{n-1}{3} \rceil) \rfloor + 1 \\ &= 1 + \lfloor \log_3(2 \cdot \lceil \frac{n-1}{3} \rceil) \rfloor + \log_3 3 \\ &= 1 + \lfloor \log_3(6 \cdot \lceil \frac{n-1}{3} \rceil) \rfloor \end{aligned} \quad (4)$$

Fall  $n$  durch 3 teilbar:  $n = 3k$

$$\begin{aligned} 1 + \lfloor \log_3(6 * \lceil \frac{3k-1}{3} \rceil) \rfloor &= 1 + \lfloor \log_3(6 * \lceil k - \frac{1}{3} \rceil) \rfloor \\ &= 1 + \lfloor \log_3(6 * k) \rfloor \\ &= 1 + \lfloor \log_3(2n) \rfloor \checkmark \end{aligned} \tag{5}$$

Fall  $n - 1$  durch 3 teilbar:  $n - 1 = 3k$

$$\begin{aligned} 1 + \lfloor \log_3(6 * \lceil \frac{3k}{3} \rceil) \rfloor &= 1 + \lfloor \log_3(6 * k) \rfloor \\ &= 1 + \lfloor \log_3(6 * \frac{n-1}{3}) \rfloor \\ &= 1 + \lfloor \log_3(2n-2) \rfloor \\ &= 1 + \lfloor \log_3(2n) \rfloor \checkmark \end{aligned} \tag{6}$$

Fall  $n + 1$  durch 3 teilbar:  $n + 1 = 3k$

$$\begin{aligned} 1 + \lfloor \log_3(6 * \lceil \frac{3k-2}{3} \rceil) \rfloor &= 1 + \lfloor \log_3(6 * \lceil k - \frac{2}{3} \rceil) \rfloor \\ &= 1 + \lfloor \log_3(6 * k) \rfloor \\ &= 1 + \lfloor \log_3(2n+2) \rfloor \\ &= 1 + \lfloor \log_3(2n) \rfloor \checkmark \end{aligned} \tag{7}$$

Damit ist gezeigt, dass die obige Gleichung  $S(n) = \lfloor \log_3(2n) \rfloor + 1$  wirklich stimmt.

□

## Aufgabe 2

### Teil a)

Wenn das gesuchte Element genau am Anfang der Liste liegt, wird in jedem Schleifendurchlauf die *if*-Bedingung in Zeile 4 *falsch* und die in Zeile 5 *wahr* sein. Der Zeiger **right** wird in jedem Schritt der Schleife um 1 nach links verschoben, bis er schließlich auch auf 0 zeigt.  $(n - 1)$ -mal wird also verschoben, und die Schleifenbedingung wird noch ein weiteres Mal geprüft, wenn beide Zeiger auf das erste Element zeigen. Daher lautet die Worst-case Gleichung:

$$W(n) = n \quad (8)$$

Da der Algorithmus quasi "symmetrisch" operiert, tritt der Worst-case auch ein, wenn das gesuchte Element ganz am Ende der Liste liegt. Der Algorithmus ließe sich verbessern, wenn er direkt abbrechen würde, sobald er das gesuchte Element findet.

### Teil b)

Für den Best-case muss das gesuchte Element in der Mitte der Liste liegen. Dann werden bei jedem Schleifendurchlauf sowohl der rechte als auch der linke Zeiger um 1 verschoben, wodurch die Suche schneller beendet wird.

#### Fall 1: $n$ ungerade

In diesem Fall gibt es ein Element, was eindeutig in der Mitte liegt, das heißt die Anzahl der Elemente rechts und links vom gesuchten Element sind gleich, nämlich  $\lfloor \frac{n}{2} \rfloor$ . Das heißt beide Zeiger müssen  $\lfloor \frac{n}{2} \rfloor - 1$  Sprünge machen, bevor sie auf die Nachbarelemente des gesuchten Elements zeigen. Nach einem weiteren Sprung zeigen beide auf das gesuchte Element. Nun muss die Schleifenbedingung noch ein weiteres Mal überprüft werden, bevor der Algorithmus fertig ist.

#### Fall 2: $n$ gerade

In diesem Fall kann das gesuchte Element nicht genau in der Mitte der Liste liegen; dort liegen zwei Elemente, die dieselbe Laufzeit besitzen. Nach  $\lfloor \frac{n}{2} \rfloor - 1$  Sprüngen zeigt der rechte Zeiger auf das *rechte* mittlere Element und der linke Zeiger auf das *linke*. Je nachdem, welches der Elemente das gesuchte ist, muss der jeweils andere Zeiger einen weiteren Sprung machen, um auch auf das Element zu zeigen. Nun muss die Schleifenbedingung noch ein weiteres Mal überprüft werden, bevor der Algorithmus fertig ist.

$$B(n) = \lfloor \frac{n}{2} \rfloor + 1 \quad (9)$$

### Teil c)

Zunächst leiten wir die allgemeine Laufzeit dafür her, dass das gesuchte Element an der Stelle  $0 \leq i < n$  ist. Unabhängig von der Position des Elements muss Zeiger A, der weiter von  $i$  entfernt ist als Zeiger B, mindestens die Hälfte der Liste durchqueren, also mindestens  $\lfloor \frac{n}{2} \rfloor$  Schritte machen. Im Best-case sind nun beide Zeiger bereits auf dem gesuchten Element, also wird nur ein weiterer Vergleich gebraucht. In allen anderen Cases muss Zeiger A aber weiter "wandern", bis er Zeiger B, der jetzt bereits auf  $i$  steht, trifft. Die Anzahl der zusätzlichen Schritte hängt vom Abstand von  $i$  zur Mitte der Liste ab, je weiter  $i$  davon entfernt ist, desto länger die Laufzeit. Der Abstand zur Mitte ist  $|\frac{n-1}{2} - i|$ ; rundet man diesen Abstand ab, so erhält man die Anzahl an zusätzlichen Sprüngen von Zeiger A. Damit ergibt sich folgende Gleichung für die Laufzeit:

$$t(n, i) = \underbrace{\lfloor \frac{n}{2} \rfloor}_{\text{Bis zur Mitte der Liste}} + \underbrace{\lfloor |\frac{n-1}{2} - i| \rfloor}_{\text{Zusatzsprünge}} + \underbrace{1}_{\text{letzter Vergleich}} \quad (10)$$

Die Wahrscheinlichkeit, dass das gesuchte Element an der Stelle  $i$  in der Liste auftaucht, ist immer  $\frac{1}{n}$ .  
Allgemein ergibt sich für den Average Case:

$$\begin{aligned} A(n) &= \sum_{I \in D_n} Pr(I) * t(I) \\ &= \sum_{i=0}^{n-1} \left[ \frac{1}{n} * t(n, i) \right] \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \left[ \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \left| \frac{n-1}{2} - i \right| \right\rfloor + 1 \right] \end{aligned} \tag{11}$$

## Aufgabe 3

### Teil a)

Zeige, dass  $(\mathbb{T}, \preceq)$  ein vollständiger Verband ist.

#### Reflexiv

Sei  $A \in \mathbb{T}$  beliebig. Es gilt:

$$\begin{aligned} & \forall n : A(n) = A(n) \\ \Rightarrow & \forall n : A(n) \leq A(n) \\ \Leftrightarrow & A \preceq A \end{aligned} \tag{12}$$

#### Transitiv

Seien  $A, B, C \in \mathbb{T}$  beliebig, so dass  $A \preceq B \preceq C$  gilt. Dann gilt:

$$\begin{aligned} & A \preceq B \preceq C \\ \Rightarrow & \forall n : A(n) \leq B(n) \leq C(n) \\ \Rightarrow & \forall n : A(n) \leq C(n) \\ \Leftrightarrow & A \preceq C \end{aligned} \tag{13}$$

#### Anti-symmetrisch

Seien  $A, B \in \mathbb{T}$  beliebig, so dass  $A \preceq B$  und  $B \preceq A$  gilt. Dann gilt:

$$\begin{aligned} & A \preceq B \wedge B \preceq A \\ \Rightarrow & \forall n : A(n) \leq B(n) \wedge B(n) \leq A(n) \\ \Rightarrow & \forall n : A(n) = B(n) \\ \Rightarrow & \forall n : A(n) \leq B(n) \\ \Rightarrow & A \preceq B \end{aligned} \tag{14}$$

□

### Teil b)

Zeige, dass  $\Phi$  monoton bezüglich  $\preceq$  ist: Es ist zu zeigen, dass gilt  $S \preceq T \Rightarrow \Phi(S) \preceq \Phi(T)$ :

$$\begin{aligned} \Phi(S) \preceq \Phi(T) & \Leftrightarrow \forall n : & \Phi(S)(n) & \leq \Phi(T)(n) \\ \Leftrightarrow \forall n : & \begin{cases} 1, & \text{falls } n = 0 \\ 2 * S(\lfloor \frac{n}{2} \rfloor) + n & \text{sonst} \end{cases} & \leq \begin{cases} 1, & \text{falls } n = 0 \\ 2 * T(\lfloor \frac{n}{2} \rfloor) + n & \text{sonst} \end{cases} \\ \Leftrightarrow \forall n > 0 : & 2 * S(\lfloor \frac{n}{2} \rfloor) + n & \leq 2 * T(\lfloor \frac{n}{2} \rfloor) + n \\ \Leftrightarrow \forall n > 0 : & S(\lfloor \frac{n}{2} \rfloor) & \leq T(\lfloor \frac{n}{2} \rfloor) \\ \Leftrightarrow \forall n : & S(n) & \leq T(n) \\ \Leftrightarrow & S & \preceq T \end{aligned} \tag{15}$$

□

## Aufgabe 4

### Teil a)

Rekursionsbaum

Höhe:  $\log_{16} n$

Breite:  $4^{\log_{16} n}$

Summe in der  $i$ -ten Zeile:  $\frac{n}{4^i}$

Summe der Blätter:  $4^{\log_{16} n} = n^{\log_{16} 4} = n^{\frac{1}{2}} = \sqrt{n}$

Damit ergibt sich

$$\begin{aligned} T(n) &= \sum_{i=0}^{\log_{16} n - 1} \frac{n}{4^i} + \sqrt{n} \\ &= n \sum_{i=0}^{\log_{16} n - 1} \left(\frac{1}{4}\right)^i + \sqrt{n} \\ &< n \sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^i + \sqrt{n} \\ &< \frac{n}{1 - \frac{1}{4}} + \sqrt{n} \\ &= \frac{4}{3}n + \sqrt{n} \in \mathcal{O}(n) \end{aligned} \tag{16}$$

**Teil b)**

Beweis durch Mastertheorem:

$$T(n) = 4 * T(\lfloor \frac{n}{16} \rfloor) + n$$

$$b = 4$$

$$c = 16$$

$$\Rightarrow E = \log_{16} 4 = \frac{1}{2}$$

$$\Rightarrow n^E = \sqrt{n}$$

$$\Rightarrow f(n) = n \in \Omega(n^{\frac{1}{2}+\epsilon}), \epsilon > 0$$

Also sind wir im 3. Fall des Mastertheorems und müssen die Nebenbedingung prüfen.

$$\begin{aligned} 4 * f\left(\frac{n}{16}\right) &\leq d * f(n) \\ 4 * \frac{n}{16} &\leq d * n \\ \frac{1}{4} &\leq d \end{aligned} \tag{17}$$

Die Nebenbedingung gilt, da es ein  $\frac{1}{4} \leq d < 1$  gibt.

Damit gilt  $T(n) \in \Theta(n)$ .

□