Prof. Dr. Christian Sohler
Nicole Funk, Dr. Matthijs Ebbens, Sebastian Zaun Sommersemester 2023

4. Übungsblatt

zur Vorlesung

Grundzüge der Informatik I

Abgabe über Ilias bis zum 3.5. 14:00 Uhr.
Besprechung in Kalenderwoche 19.

Aufgabe 1 *Binäre Suche (2 Punkte)*

Betrachten Sie das folgende Feld von Zahlen:

$$A = [6, 3, 36, 99, 2, 39, 12, 45, 23, 17, 24]$$

Strukturieren Sie die Elemente in dem Feld um, sodass Binäre Suche anwendbar ist.

Führen Sie anschließend die aus der Vorlesung bekannte Methode BinäreSuche(A,12,1,11) aus, um in dem Feld $A[1, \ldots, 11]$ nach dem Wert 12 zu suchen und geben Sie die Ausgabe, sowie die besuchten Teilfelder des Algorithmus an.

Aufgabe 2 *Teile und Herrsche (4 + 2 + 3 Punkte)*

Gegeben sei ein aufsteigend sortiertes Feld A aus n natürlichen Zahlen. Bei der Übertragung des Feldes ist ein Fehler unterlaufen. Die Diagnose des Fehlers besagt, dass es nach der Übertragung maximal einen Wert gibt, welcher verfälscht wurde. Dies bedeutet, dass die betroffene Stelle des Feldes einen größeren Wert als ursprünglich haben kann. Dadurch kann es sein, dass das Feld nach der Übertragung nicht mehr aufsteigend sortiert ist. Also kann binäre Suche auf dem Feld nicht mehr angewandt werden.

Trotz dieses Fehlers sollen aber alle unverfälschten Werte von Feld A in Laufzeit $O(\log n)$ gesucht werden können.

a) Entwickeln Sie einen Teile und Herrsche Algorithmus, welcher das Feld A, dessen Länge n und eine natürliche Zahl x bekommt, und in Laufzeit $O(\log n)$ die Position von x im Feld A zurückgibt, sofern x ein unverfälschter Wert ist. Sollte x durch den Übertragungsfehler in A entstanden sein, kann der Algorithmus in Laufzeit $O(\log n)$ die Position von x ausgeben, oder einen Fehler ausgeben.

b) Beginnen Sie die Laufzeitanalyse Ihres Algorithmus, indem Sie eine Rekursionsgleichung für die Laufzeit herleiten und angeben.

c) Beweisen Sie die Korrektheit Ihres Algorithmus.

Array sortieren:

1	2	3	4	5	6	7	8	9	10	11
2	3	6	12	17	23	24	36	39	45	99

Suche 12

$$12 \leq \overset{6}{\boxed{23}}$$

1	2	3	4	5	6
2	3	6	12	17	23

$$\overset{3}{\boxed{6}} < 12$$

4	5	6
12	17	23

$$12 \leq \overset{5}{\boxed{17}}$$

4	5
12	17

$$12 \leq \overset{4}{\boxed{12}}$$

$$\overset{4}{\boxed{12}} = 12$$

Ausgabe: 4

2.)

Binäre Suche ist nicht möglich, da auf das Pivot-Element kein Verlass ist.
Da, aber nur 1 Element verfälscht ist, können wir das Pivot Element überprüfen & gegebenenfalls einen Nachbarn verwenden.

b) Laufzeit von Find(A,n,x,l,r)

$$T(m) = \begin{cases} 3 \\ T(\frac{m}{2}) + 9 \end{cases}$$

mit $m = r - l + 1$

a)

```
FindError(A, n, x)
    return Find(A, n, x, 1, n)
```

```
Find(A, n, x, l, r)
if l ≥ r then                                                    1            1
    if A[l] = x                                                  1
        return l                                                 } 1
    else
        return "nicht vorhanden oder F"
else
    m = ⌊(l+r)/2⌋                                                              1
    if (m < n and m > 1 and A[m-1] ≤ A[m] ≤ A[m+1])                            O(1)
      or (n = 1 and A[m] ≤ A[m+1]) or (m = n and A[m-1] ≤ A[m]) then
        m = m          // m verwenden        } Wenn A[m] nicht                 1
                                             } verfälscht, mache normale
                                             } binäre Suche
    else if (m > 1 and m < n)                                                  1
        m = m+1        // m+1 verwenden                                        1
    else if m = 1      // linker Rand                                          1
        return Find(A, n, x, m+1, r)                                           T(1)
    else if m = n      // rechter Rand                                         1
        return Find(A, n, x, l, m-1)                                           T(1)
    if A[m] = x                                                                1
        return m
    else if A[m] < x                                                           }
        return Find(A, n, x, m+1, r)                                           } T(m/2)
    else                                                                       }
        return Find(A, n, x, l, m)
```

$$T(n) = \begin{cases} 3 \\ T(\frac{n}{2}) + 9 \end{cases}$$

nicht in Aufgabe gefragt.

$$\begin{aligned}
T(n) &= T(\tfrac{n}{2}) + 9 \\
&= (T(\tfrac{n}{4}) + 9) + 9 \\
&= ((T(\tfrac{n}{8}) + 9) + 9) + 9 \\
&\vdots \\
&= T(\tfrac{n}{2^h}) + \sum_{i=1}^{h} 9 \\
&= T(\tfrac{n}{2^h}) + 9h
\end{aligned}$$

$h = \log_2 n$

$$\begin{aligned}
&= T(\tfrac{n}{2^{\log_2 n}}) + 9 \cdot \log_2 n \\
&= T(\tfrac{n}{n}) + 9 \log_2 n \\
&= 3 + 9 \log_2 n \quad \in \underline{O(\log_2 n)}
\end{aligned}$$

c) Korrektheitsbeweis

Beh Find(A,n,x,ℓ,r) findet x im Teilfeld $A[\ell \dots r]$, wobei
A Länge n hat und A außer an einer Stelle aufsteigend sortiert ist

Bew per Induktion über die Länge des Teilarrays $A[\ell \dots r]$, also $m = r-\ell+1$

IA $m=1$
Da die Länge des Teilarrays 1 ist, muss $\ell=r$ gelten & somit die Bedingung in Z1 erfüllt werden.
Es wird überprüft, ob x das einzige Element ist und entsprechend zurückgegeben

IV Find(A,n,x,ℓ',r') findet x im Teilfeld $A[\ell' \dots r']$ mit $1 \leq r'-\ell'+1 < m$

IS $m>1$
Da die Länge des Teilfeldes größer 1 ist, muss $\ell \neq r$ gelten & die Bedingung in Z1 nicht erfüllt sein.
Danach wird in Z7 die mittlere Position berechnet.

Fall $A[m]$ ist nicht verfälscht
Bedingung in Z8 wird erfüllt und $A[m]$ kann als PivotElement verwendet werden. Z16-19 führen damit Binäre Suche aus.

Fall $A[m]<x$
Bedingung in Z16 wird erfüllt & Find$(A,n,x,m+1,r)$ rekursiv aufgerufen.
x befindet sich in $A[m+1..r]$, wenn es nicht verfälscht ist & laut IV findet der rekursive Aufruf es dort.

Fall $A[m]>x$
Bedingung in Z16 wird nicht erfüllt & Find(A,n,x,ℓ,m) wird in Z19 rekursiv aufgerufen
x befindet sich in $A[\ell, \dots m]$, wenn es nicht verfälscht ist & laut IV findet der rekursive Aufruf es dort

Fall $A[m]$ ist verfälscht

Fall $m=1$
Bedingung in Z12 wird erfüllt.
Da $A[m]$ verfälscht ist & davor kein Wert existiert, kann x nur in $A[2..r]$ existieren.
Der rekursive Aufruf Find$(A,n,x,\underbrace{m+1}_{2},r)$ in Z13 findet x laut IV

Fall $m=n$
Bedingung in Z14 wird erfüllt
Da $A[m]$ verfälscht ist & danach kein Wert existiert, kann x nur in $A[\ell, m-1]$ existieren.
Der rekursive Aufruf Find$(A,n,x,\ell,m-1)$ in Z15 findet x dort laut IV.

Fall $m>1$ und $m<n$

Bedingung in Z10 wird erfüllt und $m=m+1$ in Z11 gerechnet.
Da $A[m]$ verfälscht ist, kann $A[m+1]$ es nicht sein. Also wird $A[m+1]$ das neue Pivot Element.

weitere Argumentation wie bei binärer Suche.

Aufgabe 3 *Rekursionsgleichungen (3 + 3 + 3 Punkte)*

Geben Sie für die folgenden Rekursionsgleichungen eine möglichst einfache und langsam wachsene Funktion $g(n)$ an, sodass $T(n) \in O(g(n))$ ist.

Gehen Sie dabei wie folgt vor:

Lösen Sie die Rekursionsgleichungen auf, um eine nicht rekursive Funktion zu erhalten, welche die Laufzeit in Abhängikeit von n beschreibt und beweisen Sie deren Korrektheit mittels Induktion. Drücken Sie nun Ihre bewiesene Funktion durch O-Notation aus.

a) $T(n) = \begin{cases} 1 & n = 1 \\ 1 \cdot T(\frac{n}{4}) + 4 & \text{sonst} \end{cases}$

b) $T(n) = \begin{cases} 1 & n = 1 \\ 2 \cdot T(\frac{n}{2}) + n^2 & \text{sonst} \end{cases}$

c) $T(n) = \begin{cases} 1 & n = 1 \\ 3 \cdot T(\frac{n}{3}) + n & \text{sonst} \end{cases}$

3.)

a) $T(n) = \begin{cases} 1 \\ 1 \cdot T(\frac{n}{4}) + 4 \end{cases}$

$$T(n) = T(\tfrac{n}{4}) + 4$$
$$= (T(\tfrac{n}{4^2}) + 4) + 4$$
$$= (T(\tfrac{n}{4^3}) + 4) + 4 + 4$$
$$\vdots$$
$h = \log_4 n \qquad = T(\tfrac{n}{4^h}) + h \cdot 4$

$$= T(\tfrac{n}{4^{\log_4 n}}) + 4 \cdot \log_4 n$$
$$= T(\tfrac{n}{n}) + 4 \log_4 n$$
$$= 1 + 4 \log_4 n \quad \in O(\log_4 n)$$

<u>Beh</u> $T(n) \leq 1 + 4 \log_4 n$

<u>Bew</u> durch Induktion über n

<u>IA $n=1$</u>

$$T(n) = 1 = 1 + 4 \cdot 0 = 1 + 4 \cdot \log_4 1$$

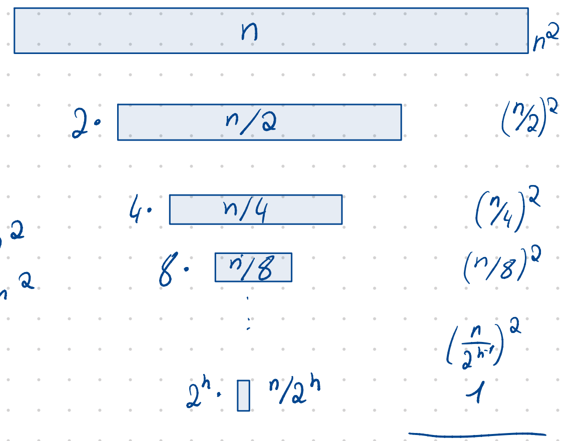
<u>IV</u> $T(k) = 1 + 4 \log_4 k \quad$ für $\quad 1 \leq k < n$

<u>IS $\frac{n}{4} \rightsquigarrow n$</u>

$$T(n) = T(\tfrac{n}{4}) + 4$$
$$\overset{IV}{=} (1 + 4 \cdot \log_4 \tfrac{n}{4}) + 4$$
$$= 5 + 4 \cdot (\log_4 n - \log_4 4)$$
$$= 5 + 4 \log_4 n - 4 \cdot 1$$
$$= 1 + 4 \log_4 n$$

b)

$$T(n) = \begin{cases} 1 \\ 2T(\frac{n}{2}) + n^2 \end{cases}$$



$$1 + \sum_{i=0}^{h-1} \left(\frac{n}{2^i}\right)^2$$

$$\begin{aligned}
T(n) &= 2T(\tfrac{n}{2}) + n^2 \\
&= 2(2T(\tfrac{n}{2^2}) + (\tfrac{n}{2})^2) + n^2 \\
&= 2(2(2T(\tfrac{n}{2^3}) + (\tfrac{n}{2^2})^2) + (\tfrac{n}{2})^2) + n^2 \\
&= 2^3 \cdot T(\tfrac{n}{2^3}) + 2^2 \cdot (\tfrac{n}{2^2})^2 + 2\cdot(\tfrac{n}{2})^2 + n^2 \\
&\vdots \\
&= 2^h T(\tfrac{n}{2^h}) + \sum_{i=0}^{h-1} 2^i (\tfrac{n}{2^i})^2 \\
&= 2^h T(\tfrac{n}{2^h}) + n^2 \cdot \sum_{i=0}^{h-1} \frac{2^i}{2^{2i}} \\
&= 2^h T(\tfrac{n}{2^h}) + n^2 \sum_{i=0}^{h-1} \frac{1}{2^i}
\end{aligned}$$

$h = \log_2 n$

$$\begin{aligned}
&= 2^{\log_2 n} T(\tfrac{n}{2^{\log_2 n}}) + n^2 \sum_{i=0}^{\log_2 n - 1} 2^{-i} \\
&= n \cdot T(\tfrac{n}{n}) + n^2 \left(2 - \tfrac{2}{n}\right) \\
&= n \cdot 1 + 2n^2 - 2n \quad = \underline{\underline{2n^2 - n \quad \in O(n^2)}}
\end{aligned}$$

<u>Beh</u> $T(n) \leq$

<u>Bew</u> per Induktion über n

<u>IA</u> $n = 1$

$$T(1) = 1 = 2 - 1 = 2 \cdot 1^2 - 1$$

<u>IV</u> $T(k) = 2k^2 - k$ für $1 \leq k < n$

<u>IS</u> $\frac{n}{2} \rightsquigarrow n$

$$\begin{aligned}
T(n) &= 2T(\tfrac{n}{2}) + n^2 \\
&\overset{IV}{=} 2(2(\tfrac{n}{2})^2 - \tfrac{n}{2}) + n^2 \\
&= 4 \cdot \frac{n^2}{2^2} - 2 \cdot \frac{n}{2} + n^2 \\
&= n^2 - n + n^2 \\
&= \underline{\underline{2n^2 - n}}
\end{aligned}$$

c)

$$T(n) = \begin{cases} 1 \\ 3 \cdot T(\frac{n}{3}) + n \end{cases}$$

$$\begin{aligned}
T(n) &= 3T(\tfrac{n}{3}) + n \\
&= 3(3T(\tfrac{n}{3^2}) + \tfrac{n}{3}) + n \\
&= 3(3(3T(\tfrac{n}{3^3}) + \tfrac{n}{3^2}) + \tfrac{n}{3}) + n \\
&= 3^3\, T(\tfrac{n}{3^3}) + n + n + n \\
&\vdots \\
&= 3^h\, T(\tfrac{n}{3^h}) + h \cdot n
\end{aligned}$$

$h = \log_3 n$

$$\begin{aligned}
&= 3^{\log_3 n}\, T(\tfrac{n}{3^{\log_3 n}}) + \log_3 n \quad n \\
&= n \cdot T(\tfrac{n}{n}) + n \log_3 n \\
&= n + n \log_3 n \quad \in \mathcal{O}(n \log_3 n)
\end{aligned}$$

Beh $T(n) \leq n + n \log_3 n$

Bew per Induktion über n

IA $n = 1$

$$T(1) = 1 = 1 + 1 \cdot 0 = 1 + 1 \cdot \log_3 1$$

IV $T(k) = k + k \log_3 k$ für $1 \leq k < n$

IS $\frac{n}{3} \rightsquigarrow n$

$$\begin{aligned}
T(n) &= 3T(\tfrac{n}{3}) + n \\
&\overset{IV}{=} 3(\tfrac{n}{3} + \tfrac{n}{3} \cdot \log_3 \tfrac{n}{3}) + n \\
&= n + n \cdot (\log_3 n - \log_3 3) + n \\
&= 2n + n(\log_3 n - 1) \\
&= n + n \log_3 n
\end{aligned}$$