

Einzelprüfung „Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft)“

Einzelprüfungsnummer 46115 / 2021 / Frühjahr

## Thema 2 / Teilaufgabe 2 / Aufgabe 3 *(Lineare und Binäre Suchverfahren)*

**Stichwörter:** Binäre Suche

Gegeben ist ein aufsteigend sortiertes Array  $A$  von  $n$  ganzen Zahlen und eine ganze Zahl  $x$ . Es wird der Algorithmus `BinarySearch` betrachtet, der  $A$  effizient nach dem Wert  $x$  absucht. Ergebnis ist der Index  $i$  mit  $x = A[i]$  oder `NIL`, falls  $x \notin A$ .

**Funktion** `BinarySearch(int A, int r)`

```
l = 1;
r = A.length;
while r ≥ l do
  if x < A[m] then
    r = m - 1;
  else if x = A[m] then
    return m;
  else
    l = m + 1;
  end
  return NIL;
end
```

- (a) Durchsuchen Sie das folgende Feld jeweils nach den in (i) bis (iii) angegebenen Werten mittels binärer Suche. Geben Sie für jede Iteration die Werte  $l, r, m$  und den betretenen if-Zweig an. Geben Sie zudem den Ergebnis-Index bzw. `NIL` an.

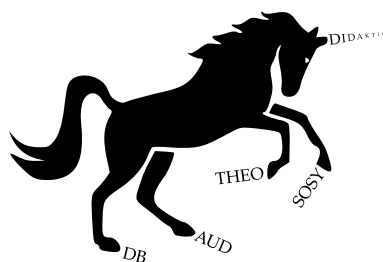
Index

i]s] «| «| 2] 4] off

wen [ilsfol7] io] w]u]al ale!

- (i) 10  
(ii) 13  
(iii) 22
- (b) Betrachten Sie auf das Array aus Teilaufgabe a). Für welche Werte durchläuft der Algorithmus nie den letzten else-Teil in Zeile 11? Hinweis: Unterscheiden Sie auch zwischen enthaltenen und nicht-enthaltenen Werten.
- (c) Wie ändert sich das Ergebnis der binären Suche, wenn im sortierten Eingabefeld zwei aufeinanderfolgende, unterschiedliche Werte vertauscht wurden? Betrachten Sie hierbei die betroffenen Werte, die anderen Feldelemente und nicht enthaltene Werte in Abhängigkeit vom Ort der Vertauschung.

- (d) Angenommen, das Eingabearray  $A$  für den Algorithmus für die binäre Suche enthält nur die Zahlen 0 und 1, aufsteigend sortiert. Zudem ist jede der beiden Zahlen mindestens ein Mal vorhanden. Ändern Sie den Algorithmus für die binäre Suche so ab, dass er den bzw. einen Index  $k$  zurückgibt, für den gilt:  $A[k] = 1$  und  $A[k-1] = 0$ .
- (e) Betrachten Sie die folgende rekursive Variante von BinarySearch.
- ```
1 int RekBinarySearch(int[] A, int x, int l, int r)
  | mi
  3 | (rekursive Implementierung)
```
- Der initiale Aufruf der rekursiven Variante lautet: `RekBinarySearch (A, z, 1, A.length)`



## Die Bschlangaul-Sammlung

### Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der  $\text{\LaTeX}$ -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/46115/2021/03/Thema-2/Teilaufgabe-2/Aufgabe-3.tex>