

## Aufgabe 1

- (a) Geben Sie für folgende Sortierverfahren jeweils zwei Felder  $A$  und  $B$  an, so dass das jeweilige Sortierverfahren angewendet auf  $A$  seine Best-Case-Laufzeit und angewendet auf  $B$  seine Worst-Case-Laufzeit erreicht. (Wir messen die Laufzeit durch die Anzahl der Vergleiche zwischen Elementen der Eingabe.) Dabei soll das Feld  $A$  die Zahlen  $1, 2, \dots, 7$  genau einmal enthalten; das Feld  $B$  ebenso. Sie bestimmen also nur die Reihenfolge der Zahlen.

Wenden Sie als Beleg für Ihre Aussagen das jeweilige Sortierverfahren auf die Felder  $A$  und  $B$  an und geben Sie nach jedem größeren Schritt des Algorithmus den Inhalt der Felder an.

Geben Sie außerdem für jedes Verfahren asymptotische Best- und Worst-Case-Laufzeit für ein Feld der Länge  $n$  an.

Die im Pseudocode verwendete Unterroutine  $\text{Swap}(A, i, j)$  vertauscht im Feld  $A$  die jeweiligen Elemente mit den Indizes  $i$  und  $j$  miteinander.

- (i) **Insertionsort**

<b>Best-Case</b>
1 2 3 4 5 6 7
<b>Worst-Case</b>
7 6 5 4 3 2 1

- (ii) Standardversion von **Quicksort** (Pseudocode s.u., Feldindizes beginnen bei 1), bei der das letzte Element eines Teilfeldes als Pivot-Element gewählt wird.
- (iii) **QuicksortVar**: Variante von Quicksort, bei der immer das mittlere Element eines Teilfeldes als Pivot-Element gewählt wird (Pseudocode s.u., nur eine Zeile neu).

Bei einem Aufruf von PartitionVar auf ein Teilfeld  $A[l \dots r]$  wird also erst mithilfe der Unterroutine  $\text{Swap } A \left[ \left\lfloor \frac{l+r-1}{2} \right\rfloor \right]$  mit  $A[r]$  vertauscht.

<b>Funktion</b> Quicksort( $A, l = 1, r = A.\text{length}$ )
<b>if</b> $l < r$ <b>then</b> $m = \text{Partition}(A, l, r);$ Quicksort( $A, l, m - 1$ ); Quicksort( $A, m + 1, r$ ); <b>end</b>

- (b) Geben Sie die asymptotische Best- und Worst-Case-Laufzeit von **Mergesort** an.

<b>Funktion</b> Partition( <i>A</i> , int <i>l</i> , int <i>r</i> )
<pre> pivot = <i>A</i>[<i>r</i>]; <i>i</i> = <i>l</i>; <b>for</b> <i>j</i> = <i>l</i> <b>to</b> <i>r</i> - 1 <b>do</b>     <b>if</b> <i>A</i>[<i>j</i>] &lt; <i>pivot</i> <b>then</b>         Swap(<i>A</i>, <i>i</i>, <i>j</i>);         <i>i</i> = <i>i</i> + 1;     <b>end</b> <b>end</b> </pre>

<b>Funktion</b> QuicksortVar( <i>A</i> , <i>l</i> = 1, <i>r</i> = <i>A</i> .length)
<pre> <b>if</b> <i>l</i> &lt; <i>r</i> <b>then</b>     <i>m</i> = PartitionVar(<i>A</i>, <i>l</i>, <i>r</i>);     QuicksortVar(<i>A</i>, <i>l</i>, <i>m</i> - 1);     QuicksortVar(<i>A</i>, <i>m</i> + 1, <i>r</i>); <b>end</b> </pre>

<b>Funktion</b> PartitionVar( <i>A</i> , int <i>l</i> , int <i>r</i> )
<pre> Swap(<i>A</i>, <math>\lfloor \frac{l+r-1}{2} \rfloor</math>, <i>r</i>); pivot = <i>A</i>[<i>r</i>]; <i>i</i> = <i>l</i>; <b>for</b> <i>j</i> = <i>l</i> <b>to</b> <i>r</i> - 1 <b>do</b>     <b>if</b> <i>A</i>[<i>j</i>] &lt; <i>pivot</i> <b>then</b>         Swap(<i>A</i>, <i>i</i>, <i>j</i>);         <i>i</i> = <i>i</i> + 1;     <b>end</b> <b>end</b> </pre>