

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

- (i) **Insertionsort**

Best-Case
1 2 3 4 5 6 7
Worst-Case
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 Unteroutine $\text{Swap } A \left[\left\lfloor \frac{l+r-1}{2} \right\rfloor \right]$ mit $A[r]$ vertauscht.

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

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

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

Funktion QuicksortVar(<i>A</i> , <i>l</i> = 1, <i>r</i> = <i>A</i> .length)
<pre> if <i>l</i> < <i>r</i> then <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>); end </pre>

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