

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

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

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

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

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