

Aufgabe 1

- (a)
- | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 3 | 6 | 9 | 4 | 2 | 1 | 8 | 5 |
| 3 | 7 | 6 | 9 | 4 | 2 | 1 | 8 | 5 |
| 3 | 4 | 6 | 9 | 7 | 2 | 1 | 8 | 5 |
| 3 | 4 | 2 | 9 | 7 | 6 | 1 | 8 | 5 |
| 3 | 4 | 2 | 1 | 7 | 6 | 9 | 8 | 5 |
| 3 | 4 | 2 | 1 | 5 | 6 | 9 | 8 | 7 |

- (b) Das Pivotelement wurde im letzten Schritt von (a) an die Position 14 gestauscht. Daher gibt $\text{PARTITION}(A, 10, 18)$ den Wert 14 zurück.

10	14					18		
3	4	2	1	5	6	9	8	7

- (c) $O(r - p)$

Aufgabe 2

- (a)
- | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|
| | 5 | 6 | 7 | 8 | | 9 | $i = 0, j = 1, L[1] \leq L[5], L[1] \leftrightarrow L[1]$ | |
| | 5 | | 6 | 7 | 8 | | 9 | $i = 1, j = 2, L[2] \leq L[5], L[2] \leftrightarrow L[2]$ |
| | 5 | 6 | | 7 | 8 | | 9 | $i = 2, j = 3, L[3] \leq L[5], L[3] \leftrightarrow L[3]$ |
| | 5 | 6 | 7 | | 8 | | 9 | $i = 3, j = 4, L[4] \leq L[5], L[4] \leftrightarrow L[4]$ |
| | 5 | 6 | 7 | 8 | | 9 | $i = 4, j = 5, L[5] \leftrightarrow L[5]$ (Pivot) | |
- \Rightarrow 5 Vertauschungen

- (b) $\text{PARTITION}(A, 1, 5)$ gibt 5 zurück.

- (c) $\text{QUICKSORT}(A, 1, 5)$

startet $\text{PARTITION}(A, 1, 5) \rightarrow 5$

startet $\text{QUICKSORT}(A, 1, 4) \rightarrow$ siehe unten

startet $\text{QUICKSORT}(A, 6, 5) \rightarrow$ Rekursion stoppt

$\text{QUICKSORT}(A, 1, 4)$

startet $\text{PARTITION}(A, 1, 4) \rightarrow 4$

startet $\text{QUICKSORT}(A, 1, 3) \rightarrow$ siehe unten

startet $\text{QUICKSORT}(A, 5, 4) \rightarrow$ Rekursion stoppt

$\text{QUICKSORT}(A, 1, 3)$

startet PARTITION($A, 1, 3$) $\rightarrow 3$
 startet QUICKSORT($A, 1, 2$) \rightarrow siehe unten
 startet QUICKSORT($A, 4, 3$) \rightarrow Rekursion stoppt

QUICKSORT($A, 1, 2$)

startet PARTITION($A, 1, 2$) $\rightarrow 2$
 startet QUICKSORT($A, 1, 1$) \rightarrow siehe unten
 startet QUICKSORT($A, 3, 2$) \rightarrow Rekursion stoppt

QUICKSORT($A, 1, 1$)

startet PARTITION($A, 1, 1$) $\rightarrow 1$
 startet QUICKSORT($A, 1, 0$) \rightarrow Rekursion stoppt
 startet QUICKSORT($A, 2, 1$) \rightarrow Rekursion stoppt

$\Rightarrow 5$ Aufrufe

Aufgabe 3

Best Case	Average Case	Worst Case
$O(n \log n)$	$O(n \log n)$	$O(n^2)$

Aufgabe 4

- *Randomized Quicksort*: Vertausche vor dem Partitionieren mit PARTITION(A, p, r) das Pivotelement an der Position r mit einem Element an einer zufällig gewählten Position $p \leq i \leq r$.
- *Median-of-three*: So lange das zu partitionierende Subarray aus mindestens drei Elementen besteht, wähle zufällig drei Elemente aus, bestimme ihren Median und vertausche ihn mit dem Element an der Pivotposition.

Aufgabe 5

	7	8	9	10	11	12	13	14	15
A	...	2	4	9	1	3	5	6	...

1. Die beiden sortierten Teile werden jeweils in einen Hilfsarray kopiert und durch einen Sentinel ∞ terminiert: $L = [2, 4, 9, \infty]$ und $R = [1, 3, 5, 6, \infty]$
2. Es werden zwei Laufvariablen i und j definiert, die jeweils auf den ersten Index von L bzw. R zeigen.
3. In einer for-Schleife von $k = 8$ bis $k = 14$ werden $L[i]$ und $R[j]$ verglichen. Wenn $L[i] \leq R[j]$, dann $A[k] \leftarrow L[i]$ und $i \leftarrow i + 1$ und sonst $A[k] \leftarrow R[j]$ und $j = j + 1$.

Aufgabe 6

12	9	17	3	8	1	21	4
9	12	17	3	8	1	21	4
9	12	3	17	8	1	21	4
3	9	12	17	8	1	21	4
3	9	12	17	1	8	21	4
3	9	12	17	1	8	4	21
3	9	12	17	1	4	8	21
1	3	4	8	9	12	17	21

Aufgabe 7

	Speicherbedarf	stabil
Quicksort	n	nein
Mergesort	$2n$	ja

Aufgabe 8

$$A = [4, 2, 0, 4, 3, 2, 4, 3, 2]$$

- Erzeuge $C = [0, 0, 0, 0, 0]$
- Erzeuge $B = [-, -, -, -, -, -, -, -, -]$ (Ergebnis)
- Zähle die Elemente von A mit C : $C = [1, 0, 3, 2, 3]$
- Kumuliere die Elemente in C : $C = [1, 1, 4, 6, 9]$

C besagt, dass

- die Zahl 0 den ersten Platz
- die Zahl 1 keinen Platz
- die Zahl 2 die Plätze 2-4

...

im Ergebnisarray B beanspruchen.

- Durchlaufe A rückwärts ($k = 9, 8, \dots, 1$)
 - $A[9] = 2$ (welches Element?)
 - $C[2] = 4$ (letzter Platz von 2 in B)
 - $B[4] \leftarrow 2$ (Element am richtigen Platz ablegen)
 - $B = [-, -, -, \mathbf{2}, -, -, -, -, -]$
 - $C[2] \leftarrow C[2] - 1$ (damit die nächste 2 *vor* dieser 2 steht)
 - $C = [1, 1, \mathbf{3}, 6, 9]$

Aufgabe 9

- (a) Voraussetzungen: Die n Elemente des Arrays müssen *natürliche Zahlen* aus einem *beschränkten* Intervall mit k Werten sein (oder sich eineindeutig darauf abbilden lassen).
- (b) Laufzeitkomplexität: $O(n + k)$ bzw. $O(n)$, da $k \leq n$
- (c) Speicherbedarf: $n + k$ (zusätzlich) sonst $2n + k$
- (d) stabil: ja