

---

# Sortieralgorithmen

## Übungen

---

1. Juni 2025

### Aufgabe 1

Sortiere die Zahlen im Array  $A = [8, 3, 7, 6, 2, 4]$  schrittweise mit Selectionsort in aufsteigender Reihenfolge. Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?


### Aufgabe 2

Sortiere die Zahlen im Array  $A = [9, 3, 5, 2, 1, 4, 8]$  schrittweise mit Selectionsort in aufsteigender Reihenfolge. Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?


### Aufgabe 3

Sortiere die Zahlen im Array  $A = [8, 3, 7, 6, 2, 4]$  schrittweise mit Insertionsort in aufsteigender Reihenfolge. Wie viele Vergleiche und Verschiebungen sind dafür insgesamt nötig?


### Aufgabe 4

Sortiere die Zahlen im Array  $A = [3, 1, 4, 2, 7, 9]$  schrittweise mit Insertionsort in aufsteigender Reihenfolge. Wie viele Vergleiche und Verschiebungen sind dafür insgesamt nötig?


### Aufgabe 5

Sortiere die Zahlen im Array  $A = [3, 1, 4, 2, 7, 9]$  schrittweise mit Bubblesort in aufsteigender Reihenfolge. Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?


### Aufgabe 6

Sortiere die Zahlen im Array  $A = [7, 5, 4, 1]$  schrittweise mit Bubblesort in aufsteigender Reihenfolge. Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?


## Aufgabe 7

- (a) Zeige schrittweise, wie die Funktion PARTITION im Quicksort-Algorithmus die unten gegebene Teiliste verarbeitet. Notiere nur jeden neuen Zustand der Liste.
- (b) Welchen Wert gibt die Funktion PARTITION in (a) für  $p = 10$  zurück?

$p$									$r$
7	3	6	9	4	2	1	8	5	

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--



### Aufgabe 9

Sortiere die Zahlen im Array  $A = [2, 4, 7, 1]$  schrittweise mit Quicksort in aufsteigender Reihenfolge.

Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?


### Aufgabe 10

Bestimme die

- (a) minimale Rekursionstiefe
- (b) maximale Rekursionstiefe

die das Quicksort-Verfahren bei einem Array mit 100 Elementen bewältigen muss, bis die zu verarbeitenden Teilarrays die Länge 1 haben.

### Aufgabe 11

Nenne zwei Modifikationen, mit denen die Worst Case-Laufzeit des Quicksort-Verfahrens vermieden werden kann.

### Aufgabe 12

Beschreibe zwei Situationen, in denen man jeweils Quicksort bzw. Mergesort bevorzugt anwenden sollte.

### Aufgabe 13

Sortiere die Zahlen im Array  $A = [7, 3, 8, 5, 9, 2, 4, 1]$  schrittweise mit der iterativen Version von Mergesort in aufsteigender Reihenfolge.


### Aufgabe 14

Sortiere die Zahlen im Array  $A = [6, 5, 8, 9, 3, 6, 0, 7, 4, 2]$  schrittweise mit der iterativen Version von Mergesort in aufsteigender Reihenfolge.


### Aufgabe 15

Zeige, wie die Zahlen im Array  $A = [3, 2, 1, 1, 0, 1, 3, 2]$  mit Countingsort aufsteigend sortiert werden, indem du die dafür nötigen Arrays angibst.

### Aufgabe 16

Zeige, wie die Zahlen im Array  $A = [2, 5, 3, 2, 2, 3, 2, 5, 1, 2]$  mit Countingsort aufsteigend sortiert werden, indem du die dafür nötigen Arrays angibst.

### Aufgabe 17

Gib die Worst- und Best-Case Laufzeiten der folgenden Sortieralgorithmen in Abhängigkeit der Arraylänge  $n$  an.

	Worst Case	Best Case
Mergesort		
Selectionsort		
Countingsort		
Insertionsort		
Quicksort		

