

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

Sortiere die Zahlen im Array $A = [8, 1, 3, 2, 7, 9, 4]$ schrittweise mit Quicksort in aufsteigender Reihenfolge.

Wie viele Vergleiche und Vertauschungen sind dafür insgesamt nötig?

Aufgabe 8

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 9

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

Wie viele Vergleiche und Verschiebungen sind dafür insgesamt nötig?

Aufgabe 10

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

Wie viele Vergleiche und Verschiebungen sind dafür insgesamt nötig?

Aufgabe 11

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 12

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 13

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 14

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

Aufgabe 15

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		

Aufgabe 16

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

Aufgabe 17

Eine Implementierung für Mergesort benötigt für das Sortieren von 10^6 Zahlen 3.6 Sekunden. Wie lange benötigt diese Implementierung für das Sortieren von 10^7 Zahlen?

Aufgabe 18

Schreibe eine Python-Funktion `isSorted(A)`, deren formales Argument `A` ein Array aus Zahlen ist und die den Wert `True` ausgibt, falls das Array bereits aufsteigend sortiert ist und `False` sonst.