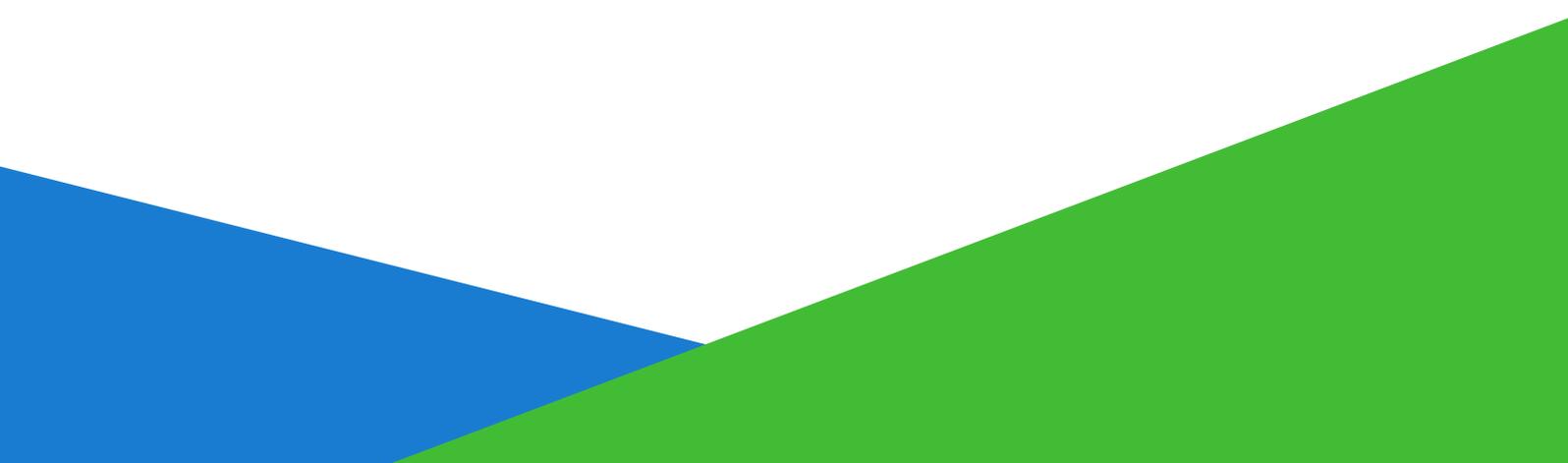


Lehreinheit mit Kopiervorlagen

## **Sortierverfahren**

Wie Datenfelder sortiert werden können und wie sich die Laufzeiten der Verfahren unterscheiden



Dr. Daniel Scholz  
Braunschweig  
daniel@taralino.de

## Herausgeber

taralino e. V.  
Braunschweig  
post@taralino.de  
www.taralino.de



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung, Nicht-kommerziell, Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-NC-SA 4.0).

<https://creativecommons.org/licenses/by-nc-sa/4.0>

Dies bedeutet insbesondere:

- Das Material darf in beliebigen Formaten (z.B. ausgedruckt oder als digitale Version) uneingeschränkt weiterverbreitet werden.
- Das Material darf verändert und weiterentwickelt werden. Ist anschließend eine erneute Veröffentlichung geplant, muss dafür jedoch auch eine Creative Commons Lizenz verwendet werden.
- Eine Nutzung des Materials für kommerzielle Zwecke ist nicht gestattet.

Stand des vorliegenden Dokuments: 23. Januar 2023

# Vorwort

Der Fokus dieser Lehrinheit liegt auf der Erarbeitung ausgewählter Algorithmen, die dazu dienen, ein Datenfeld von Zahlen zu sortieren.

Zunächst soll in einer Gruppenarbeit experimentell herausgefunden werden, wie die Arbeitsweise eines Sortierverfahrens gestaltet werden könnte. Anschließend wird das Bubblesort-Verfahren im Detail besprochen sowie ein zugehöriger Quellcode diskutiert. Darüber hinaus wird das Mergesort-Verfahren als rekursiven Algorithmus vorgestellt, um insbesondere die Laufzeit zwischen Bubblesort und Mergesort beim Sortieren großer Datenfelder zu vergleichen.

- Zielgruppe: Schulklassen ab Jahrgang 9
- Zeitaufwand: etwa 2-4 Schulstunden
- Material: Scheren sowie PC oder Tablet mit Internetzugang

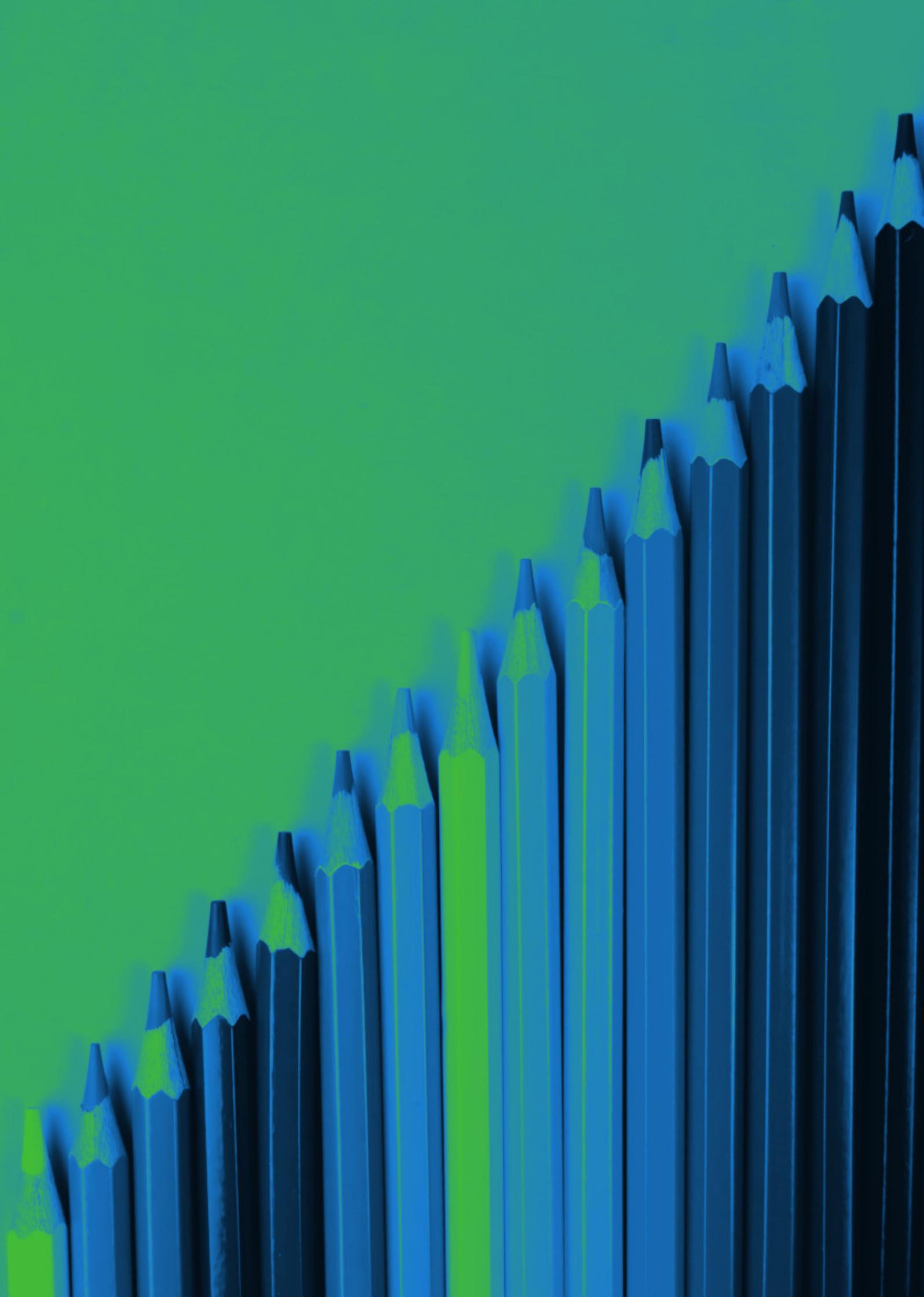
2  
/  
3

Neben der Sortierverfahren wird in dieser Lehrinheit vor allem Wert darauf gelegt, die algorithmische Problemlösekompetenz auszubauen. Hierzu dient insbesondere die Gruppenarbeit, in der eine eigene algorithmische Vorgehensweise zum Sortieren entwickelt werden soll. Zudem ist es wichtig zu verstehen, dass es sehr unterschiedliche Algorithmen geben kann, die ein exakt identisches Ergebnis liefern (hier ein sortiertes Datenfeld). Dennoch haben unterschiedliche Algorithmen ihre Vor- und Nachteile. Die wichtigsten Kriterien sind die Laufzeit oder Komplexität des Verfahrens sowie der benötigte Speicherbedarf.

Viel Spaß beim Lernen und Lehren

Daniel Scholz

September 2022



# Inhaltsverzeichnis

<b>1</b>	Gruppenarbeit . . . . .	7
<b>2</b>	Bubblesort . . . . .	10
<b>3</b>	Mergesort . . . . .	13
<b>4</b>	Laufzeitanalyse . . . . .	16

5

14

2

23

42

-4

21

4

# 1

## Gruppenarbeit

Die Lehreinheit beginnt, indem Kleingruppen gebildet werden. Jeder Gruppe wird anschließend eine Zettel mit acht oder 16 Zahlenkarten überreicht. Nun werden folgende Aufgaben gestellt:

1. Schneidet die Zahlenkarten aus und legt sie **umgedreht** auf den Tisch.
2. Mischt die Karten und legt sie anschließend in einer Reihe vor euch hin.
3. Eure Aufgabe lautet, die Zahlenkarten zu sortieren. Dazu können einzelne Karten aufgedeckt werden, aber es dürfen nur maximal **zwei Karten** gleichzeitig aufgedeckt sein. Die Reihenfolge der Karten kann jederzeit beliebig verändert werden.
4. Überlegt euch eine Vorgehensweise, um die Aufgabe zu lösen.
5. Führt die Vorgehensweise durch und überprüft euer Ergebnis, indem am Ende alle Karten aufgedeckt werden.

6  
/  
7

Es sei darauf hingewiesen, dass es nicht vorgesehen ist, sich die Zahlen umgedrehter Karten zu merken.



Falls die Teilnehmenden bereits mit Struktogrammen vertraut sind, kann folgende Zusatzaufgabe gestellt werden:

Entwerft ein Struktogramm, das eure entwickelte Vorgehensweise algorithmisch beschreibt.

Im Anschluss an die Arbeitsphase stellen die Gruppen ihre jeweiligen Ergebnisse vor. Es kann darüber diskutiert werden, ob die vorgeschlagene Vorgehensweise stets die Aufgabe löst oder ob es zu Problemen kommen kann. Es kann auch darüber gesprochen werden, dass aus der Aufgabenstellung nicht hervorgeht, ob die Zahlen auf- oder absteigend sortiert werden sollen. Zudem kann bereits auf die Komplexität sowie die Laufzeit der unterschiedlichen Lösungsansätze eingegangen werden.

Abschließend sollte der Lehrende anhand der Zahlenkarten das Bubblesort-Verfahren vorstellen.

Hinweis: Anhand der Zahlenkarten kann auch das Mergesort-Verfahren vorgestellt werden. Dies kann und sollte zu einem späteren Zeitpunkt der Lehrinheit vorgenommen werden.

# 2

## Bubblesort

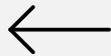
Ein vergleichsweise einfaches Sortierverfahren, welches in der Regel bereits anhand der Zahlenkarten besprochen wurde, ist das Bubblesort-Verfahren: Ein zu sortierendes Datenfeld wird von links nach rechts durchlaufen. Dabei wird in jedem Schritt das aktuelle Element mit dem rechten Nachbarn verglichen. Falls das linke Element größer ist als das rechte Element, werden die Werte der beiden Elemente getauscht. Am Ende des ersten Durchgangs steht das größte Element am Ende des Datenfeldes. Nach dem zweiten Durchgang hat das vorletzte Element des Datenfeldes den zweitgrößten Wert des gesamten Datenfeldes und so weiter.

t

Um die Vorgehensweise des Verfahrens im Detail nachzuvollziehen, dient das Arbeitsblatt auf der nachfolgenden Seite. Zudem präsentiert folgende Anwendung eine Animation des Bubblesort-Verfahrens:

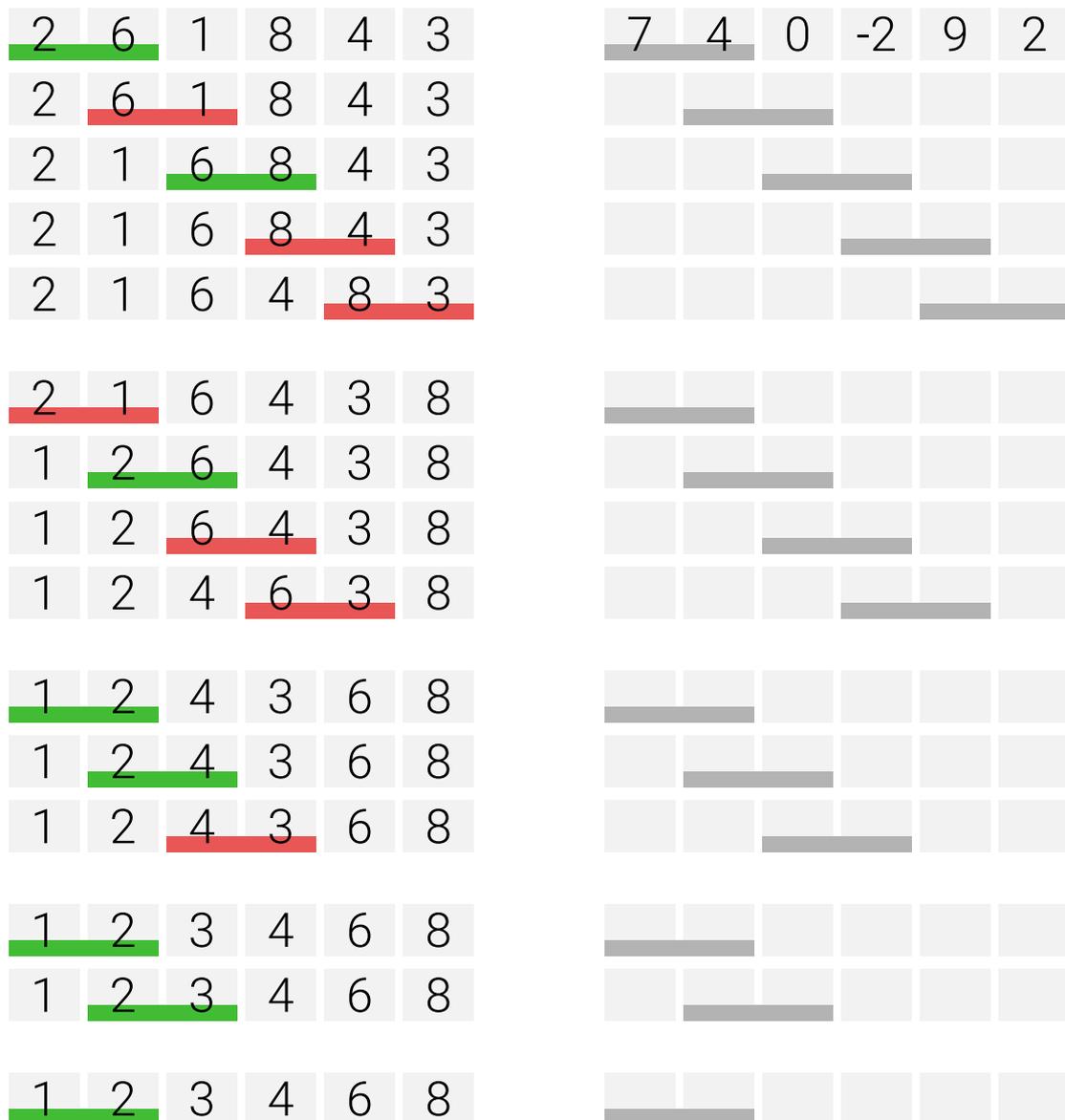


[www.taralino.de/view/M87SCWRS](http://www.taralino.de/view/M87SCWRS)



# Bubblesort

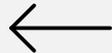
Mache dich mit der Durchführung des Bubblesort-Verfahrens anhand des linken Beispiels vertraut. Führe das Verfahren anschließend anhand des rechten Beispiels durch.



Je nach Programmiererfahrung kann anschließend die Aufgabe gestellt werden, das Bubblesort-Verfahren zu implementieren. Dazu kann den Schülerinnen und Schülern folgende Vorlage zur Verfügung gestellt werden:



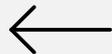
[www.taralino.de/view/UK398P22](http://www.taralino.de/view/UK398P22)



Eine Art Musterlösung lässt sich hier abrufen:



[www.taralino.de/view/UC5M4ZEC](http://www.taralino.de/view/UC5M4ZEC)



Dabei wird das Bubblesort-Verfahren bewusst in einer Funktion ausgeführt. Das zu sortierende Datenfeld wird zunächst kopiert, damit es während des Bubblesort-Verfahrens nicht verändert wird. Dieser Schritt ist nicht zwingend notwendig, macht aber später den Vergleich zum Mergesort einfacher. Schließlich wird das neue Datenfeld sortiert und zurückgegeben.

# 3

## Mergesort

Das Bubblesort-Verfahren lässt sich vergleichsweise einfach verstehen und implementieren. Es hat jedoch den Nachteil, dass es verhältnismäßig lange dauert, große Datenfelder zu sortieren (mehr dazu im nachfolgenden Abschnitt).

In der Praxis werden daher Sortierverfahren eingesetzt, die ein identisches Ergebnis liefern, jedoch deutlich weniger Laufzeit in Anspruch nehmen. Einer dieser Algorithmen ist das Mergesort-Verfahren. Dabei wird das zu sortierende Datenfeld zunächst in viele kleine Datenfelder zerlegt, die jeweils nur ein Element enthalten. Anschließend werden in einer Art Reißverschlussverfahren je zwei sortierte Datenfelder zu einem größeren stets sortierten Datenfeld zusammengefügt.

Wie genau die Arbeitsweise des Mergesort-Verfahrens ist, kann anhand der folgenden beiden Aufgabenblätter erarbeitet werden.

12  
/  
13

An dieser Stelle lassen sich wieder die Zahlenkarten einsetzen, um das Mergesort-Verfahren durch die Lehrkraft zu demonstrieren.

Das Mergesort-Verfahren ist etwas aufwendiger zu programmieren. Es kann jedoch besonders elegant implementiert werden, indem rekursive Funktionen eingesetzt werden:

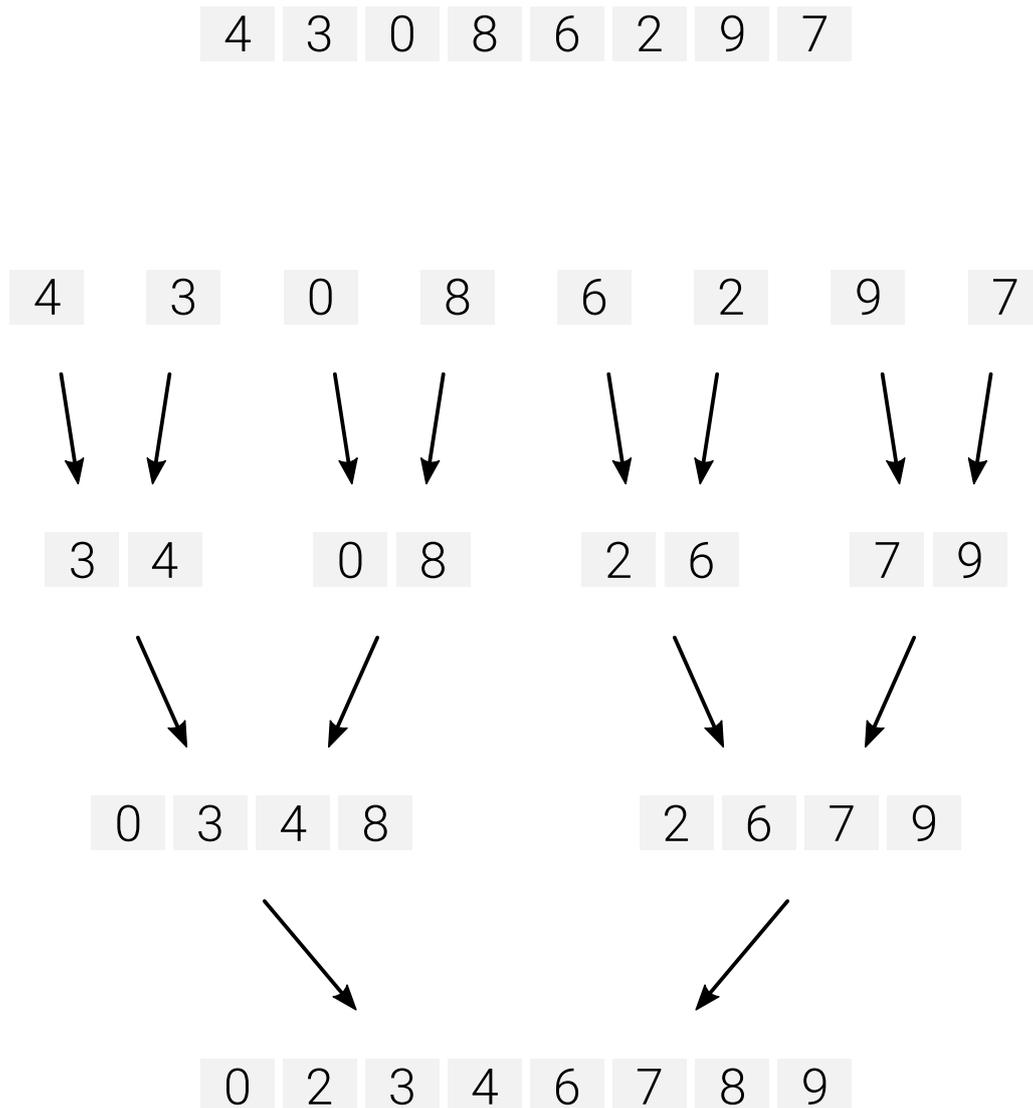


[www.taralino.de/view/L36MZKPL](http://www.taralino.de/view/L36MZKPL)



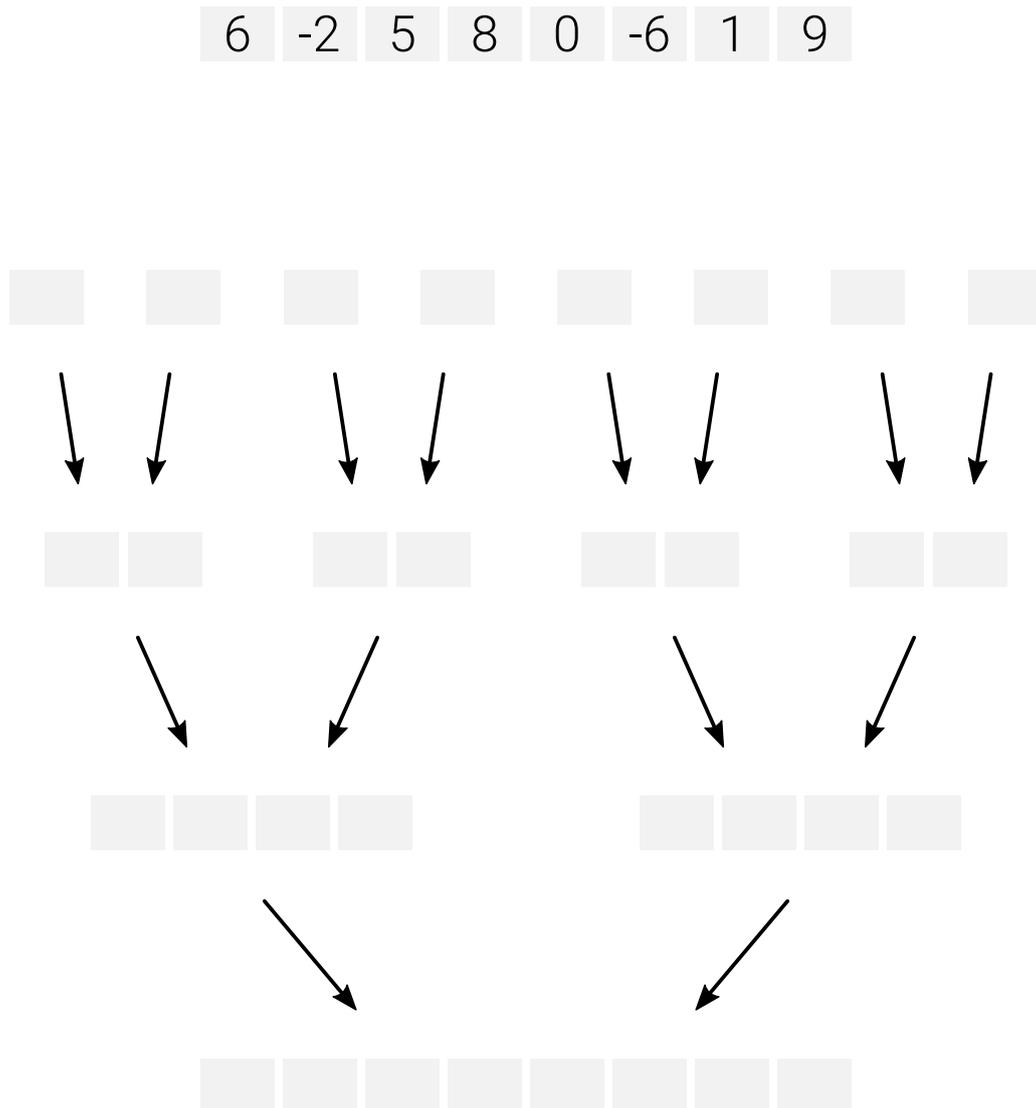
# Mergesort

Mache dich mit der Durchführung des Mergesort-Verfahrens anhand des Beispiels vertraut.



# Mergesort

Führe das Mergesort-Verfahren anhand des folgenden Beispiels durch.



# 4

## Laufzeitanalyse

Mit Bubblesort und Mergesort wurden zwei Sortierverfahren vorgestellt, die beide die gleiche Aufgabe lösen und ein identisches Ergebnis liefern.

Ein wichtiges Qualitätsmerkmal, welches daher nun untersucht werden soll, ist die Komplexität der beiden Algorithmen im Vergleich. Genauer geht es darum, die Laufzeit der Algorithmen in Abhängigkeit der Anzahl der zu sortierenden Zahlen zu untersuchen, siehe nachfolgendes Arbeitsblatt.

t

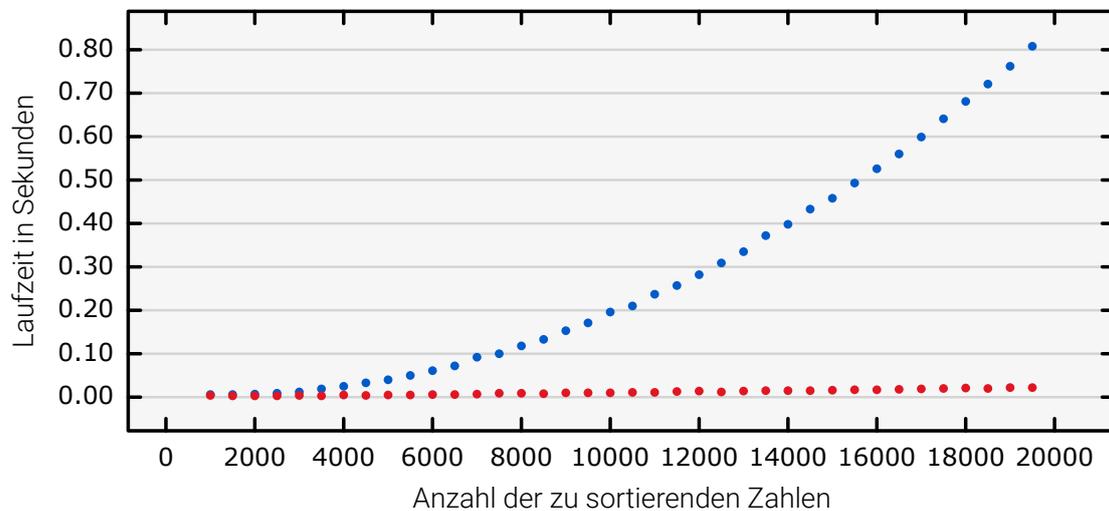


Abb. 1 Laufzeitanalyse im Vergleich zwischen Bubblesort (blau) und Mergesort (rot)

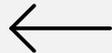
Die Ergebnisse des Arbeitsblattes können auch in eine grafische Darstellung übertragen werden, vgl. Abb. 1.

# Laufzeitanalyse

Unter folgendem Link findest du sowohl das Bubblesort- als auch das Mergesort-Verfahren.



[www.taralino.de/view/PFJKWSJM](http://www.taralino.de/view/PFJKWSJM)



Verändere jeweils im Quellcode die Anzahl der Elemente des zu sortierenden Datenfeldes, indem der Wert der Variablen  $n$  angepasst wird. Führe damit eine Laufzeitanalyse gemäß der folgenden Tabelle durch.

Größe des Datenfeldes (Anzahl der Elemente)	5.000	10.000	20.000	30.000	40.000
Laufzeit Bubblesort in Sekunden					
Laufzeit Mergesort in Sekunden					
Größe des Datenfeldes (Anzahl der Elemente)	50.000	60.000	70.000	80.000	90.000
Laufzeit Bubblesort in Sekunden					
Laufzeit Mergesort in Sekunden					

Nach der Bearbeitung des Arbeitsblattes zur Laufzeitanalyse sollten die Schülerinnen und Schülern eindrucksvoll erlebt haben, dass das Bubblesort-Verfahren zum Sortieren sehr großer Datenfelder aufgrund der Laufzeit ungeeignet ist.

Tatsächlich besitzt der Bubblesort-Algorithmus eine Komplexität von

$$\mathcal{O}(n^2),$$

wobei  $n$  die Anzahl der zu sortierenden Zahlen ist. Dies hat zur Folge, dass sich die Laufzeit vervierfacht, falls sich  $n$  verdoppelt. Die Laufzeit verhundertfacht sich, falls sich die Anzahl der zu sortierenden Zahlen verzehnfacht.

Die Komplexität des Mergesort-Verfahrens ist

$$\mathcal{O}(n \cdot \log(n)).$$

Damit ist die Komplexität nicht ganz, aber fast linear: Falls sich  $n$  verdoppelt, so wird sich auch die Laufzeit des Verfahrens kaum mehr als verdoppeln.



# Arbeitsblatt

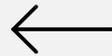
Informiere dich im Internet über die Arbeitsweise des Selectionsort-Verfahrens:

1. Beschreibe die Unterschiede zwischen Bubblesort und Selectionsort.
2. Erprobe das Selectionsort-Verfahren anhand der Zahlenkarten wie zuvor (es dürfen maximal zwei Karten gleichzeitig aufgedeckt sein).

Implementiere das Selectionsort-Verfahren, wobei du folgende Vorlage verwenden kannst:



[www.taralino.de/view/UK398P22](http://www.taralino.de/view/UK398P22)



# Bildernachweise

Die folgende Liste dient als Nachweis über verwendete Fotomaterial (in ihrer jeweiligen Originalversion):

Seite 4: Photo by *Tamanna Rume* on Unsplash

