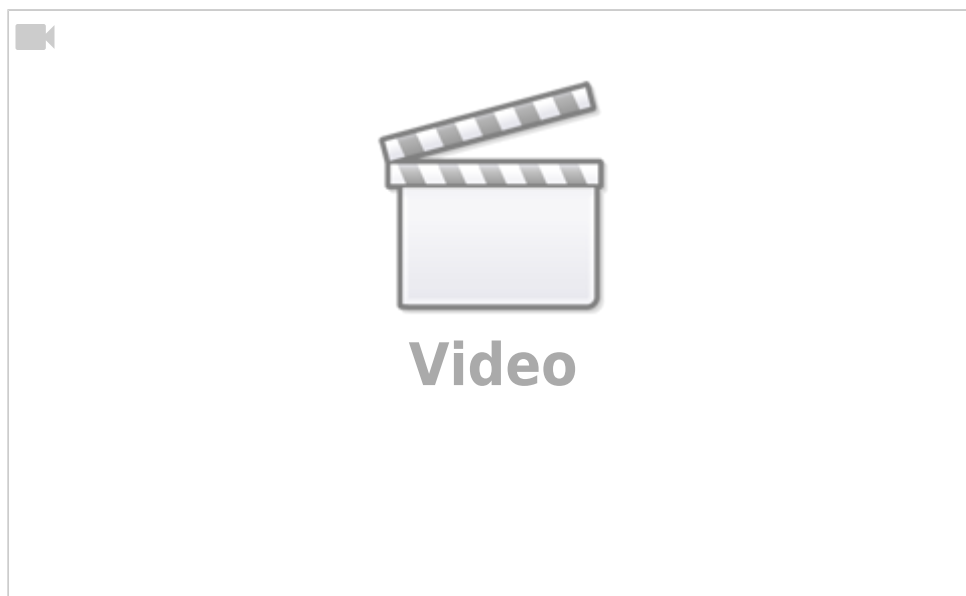


# Quicksort - effizienter sortieren

Wenn du diesen Abschnitt nicht lösen oder bearbeiten kannst, ist das nicht schlimm. Schau dir dieses Video bis zum Zeitindex 5:00 an.



Es wird dir ein weiterer Sortieralgorithmus mit dem Namen Quicksort angezeigt. Es wird die gleiche Zahlenfolge [hier](#) sortiert.

- Könntest du das Verfahren in einer Gruppe durchspielen wie bei Bubblesort?
- Was sagt dein Gefühl über die Anzahl der Vergleiche, die für die Sortierung benötigt werden, im Unterschied zu Bubblesort?

## Auflösung

Das Aufwändige für einen Rechner sind Vergleiche. Im Prinzip muss man für einen Vergleich eine Subtraktion durchführen und dann zwei Fälle unterscheiden:

1. Ist das Ergebnis 0, dann sind die Zahlen gleich
2. Ist das Ergebnis von 0 verschieden, dann sind die Zahlen nicht gleich

Ein Tausch von Zahlen ist dagegen vergleichsweise einfach - es wird intern nur ein Zeiger umgestellt. Bei [Bubblesort](#) hast du gesehen, dass für die gegebene Zahlenreihe 40 Vergleiche notwendig sind, um den Algorithmus abzuschließen.

Quicksort ist etwas schwieriger syntaktisch aufzuschreiben, aber wir schauen mal.

- das aktuelle Pivotelement ist **fett** formatiert.
- eine unterstrichene Zahl ist ein Zahl, die mit dem Pivot-Element verglichen wurde
- die Anzahl der unterstrichenen Zahlen ist also die Gesamtzahl der Vergleiche im aktuellen Durchgang

### Durchgang 1 - 6 ist Pivotelement:

3 7 1 8 2 5 9 **4** **6**

3 4 1 8 2 5 9 7 **6** ( 4 und 7 wurden getauscht)

3 4 1 8 2 5 9 7 **6**

3 4 1 5 2 8 9 7 **6** ( 5 und 8 wurden getauscht)

[ 3 4 1 5 2 ] 6 [ 9 7 8 ] ( 6 und 8 wurden getauscht, Vergleich war unnötig, die 6 muss die kleinste Zahl sein)

Wir haben in diesem ersten Durchgang insgesamt 7x Zahlen miteinander verglichen. Die 6 liegt jetzt schon auf ihrer Endposition.

Wir werden weiterhin ...

- für das Pivot-Element **2** 3x vergleichen müssen
- für das Pivot-Element **4** 2x vergleichen müssen
- für das Pivot-Element **8** 1x vergleichen müssen

Wir kommen also mit 13 Vergleichen aus. Quicksort ist um Größenordnungen effizienter als Bubblesort. Bei einer bereits sortierten Liste (Best-Case) bringt Quicksort keinen Nachteil gegenüber Bubblesort.

From:

<https://schule.riecken.de/> - Unterrichtswiki

Permanent link:

<https://schule.riecken.de/doku.php?id=informatik:lesson:klasse10:algorithmisch:quicksort>

Last update: **2024/08/08 08:04**

