

Quick-Sort in Java

➔ Rekursiver Sortieralgorithmus, nutzt eine zentrale Zahl (eng. Pivot) je Funktionsausführung.

Gegeben sei ein zufälliges Array als Input:

2	3	9	4	3	8	11	5
---	---	---	---	---	---	----	---

1. Pivot muss bestimmt werden: Median aus den **Zahlen** der ersten (Index = 0), mittleren (Index = Array-Länge / 2) und letzten Stelle des Arrays; hier **4**. Diese drei **Zahlen** werden von klein nach groß sortiert. Wenn das Array nur aus maximal drei Zahlen besteht, wird hier abgebrochen.

2	3	9	4	3	8	11	5
---	---	---	---	---	---	----	---

2. Pivot muss mit der Zahl an letzter Stelle des Arrays vertauscht werden:

2	3	9	5	3	8	11	4
---	---	---	---	---	---	----	---

3. Schrittweise werden **Zahlen von links** größer als der **Pivot** und **Zahlen von rechts** kleiner als der **Pivot** betrachtet. Diese Zahlen werden vertauscht:

2	3	9	5	3	8	11	4
2	3	3	5	9	8	11	4

4. Wenn der Index der **Zahl von links** größer ist als der Index der **Zahl von rechts**, wird abgebrochen und der **Pivot** mit der aktuellen **Zahl von Links** getauscht:

2	3	3	5	9	8	11	4
2	3	3	4	9	8	11	5

Nun sind alle Zahlen vor dem Pivot kleiner und alle Zahlen nach dem Pivot größer als der Pivot. Die Zahlen sind in sich aber noch nicht nach Größe geordnet. Hier findet die Rekursion statt:

5. Das Array wird in zwei kleinere Arrays unterteilt; Ein Array mit den Zahlen links des Pivots und ein Array mit den Zahlen rechts des Pivots (der Pivot selbst steht schon an der richtigen Stelle). Mit den zwei neuen Arrays werden die Schritte von 1. an jeweils von vorn ausgeführt. Die zwei neuen Arrays werden anschließend auch solange in 2 Sub-Arrays unterteilt, bis der Array so klein ist, dass das Programm an Punkt 1 abbricht.
6. Die Sub-Arrays werden in den Array eingefügt und dieser wird ausgegeben.