

# Selectionsort: Sortieren durch Auswählen<sup>1</sup>

## Weiterführende Literatur:

- *Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 39*
- *Wikipedia-Artikel „Selectionsort“*
- *Saake und Sattler, Algorithmen und Datenstrukturen, Seite 127-129 (PDF 145-147)*
- *Schneider, Taschenbuch der Informatik, 6.4.1 Naive Sortierverfahren, Seite 191*

- Funktionsweise
  - solange zu sortierende Liste mehr als ein Element beinhaltet:
    - lösche das *Maximum* / *Minimum* aus der Liste
    - füge es ans Ende der Ergebnisliste
    - wiederhole, bis Eingangsliste leer
- *Eigenschaften* von Selectionsort:
  - Laufzeitkomplexität:  $\mathcal{O}(n^2)$  (im *Best*-, *Average*- und *Worst-Case*)
  - Stabilität leicht erreichbar
  - bei Zahlen *in-situ*

## Minimale zum Auswendiglernen

```
11  /**
12   * Sortiere mit Hilfe des Insertionsort-Algorithmus.
13   *
14   * <p>
15   * <strong>Abkürzungen:</strong>
16   * </p>
17   *
18   * <ul>
19   * <li>m: markierung
20   * </li>
21   */
22  public int[] sortiere() {
23      int m = a.length - 1;
24      while (m >= 0) {
25          int max = 0;
26          for (int i = 1; i <= m; i++) {
27              if (a[i] > a[max]) {
28                  max = i;
29              }
30          }
31          vertausche(m, max);
32          m--;
33      }
34      return a;
35  }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/beschlangaul/sortier/SelectionMinimal.java](https://github.com/beschlangaul/sortier/SelectionMinimal.java)

<sup>1</sup>Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 39.

## Iterativ (Auswahl von rechts nach links)

```

11 public int[] sortiere() {
12     // Am Anfang ist die Markierung das letzte Element im Zahlen-Array.
13     int markierung = zahlen.length - 1;
14     while (markierung >= 0) {
15         berichte.feldMarkierung(markierung);
16         // Bestimme das größtes Element.
17         // max ist der Index des größten Elements.
18         int max = 0;
19         // Wir vergleichen zuerst die Zahlen mit der Index-Number
20         // 0 und 1, dann 1 und 2, etc. bis zur Markierung
21         for (int i = 1; i <= markierung; i++) {
22             if (zahlen[i] > zahlen[max]) {
23                 max = i;
24             }
25         }
26
27         // Tausche zahlen[markierung] mit dem gefundenem Element.
28         vertausche(markierung, max);
29         // Die Markierung um eins nach vorne verlegen.
30         markierung--;
31     }
32     return zahlen;

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/sortier/SelectionRechtsIterativ.java](https://github.com/bschlangaul/sortier/SelectionRechtsIterativ.java)

2

## Iterativ (Auswahl von links nach rechts)

```

10 public int[] sortiere() {
11     int markierung = 0;
12     while (markierung < zahlen.length - 1) {
13         int min = markierung;
14         for (int i = markierung; i < zahlen.length; i++) {
15             if (zahlen[i] < zahlen[min]) {
16                 min = i;
17             }
18         }
19         vertausche(markierung, min);
20         markierung++;
21     }
22     return zahlen;
23 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/sortier/SelectionLinksIterativ.java](https://github.com/bschlangaul/sortier/SelectionLinksIterativ.java)

## Halbrekursiv

```

15 private void sortiereHalbRekursiv(int index, int anzahl) {
16     if (anzahl <= 0) {
17         return;
18     }
19
20     int minimum = index;

```

<sup>2</sup>Saake und Sattler, *Algorithmen und Datenstrukturen*, Seite 128 (PDF 146).

```

21     for (int i = index + 1; i < anzahl; i++) {
22         if (zahlen[i] < zahlen[minimum]) {
23             minimum = i;
24         }
25     }
26     vertausche(minimum, index);
27     if (index + 1 < anzahl) {
28         sortiereHalbRekursiv(index + 1, anzahl);
29     }
30 }
31
32 public int[] sortiere() {
33     sortiereHalbRekursiv(0, zahlen.length);
34     return zahlen;
35 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/sortier/SelectionHalbRekursiv.java](https://github.com/src/main/java/org/bschlangaul/sortier/SelectionHalbRekursiv.java)

## Rekursiv

```

9     /**
10      * Gib die Indexposition im Zahlenfeld zurück, das mit der kleinsten Zahl
11      ↪ belegt
12      * ist.
13      *
14      * @param i Die Index-Nummer der einen Zahlen, mit der verglichen werden soll.
15      * @param j Die Index-Nummer der anderen Zahlen, mit der verglichen werden
16      ↪ soll.
17      *
18      * @return Die Indexposition im Zahlenfeld zurück, das mit der kleinsten Zahl
19      *         belegt.
20      */
21     private int gibMinIndex(int i, int j) {
22         if (i == j) {
23             return i;
24         }
25         int k = gibMinIndex(i + 1, j);
26         return (zahlen[i] < zahlen[k]) ? i : k;
27     }
28
29     /**
30      * Sortiere ein Zahlen-Feld mit Hilfe des Selectionsort-Algorithmus in einer
31      * rekursiven Art und Weise.
32      *
33      * @param index Ab welcher Index-Nummer im Zahlen-Feld sortiert werden soll.
34      * @param anzahl Wie viele Zahlen aber der gegebenen Index-Nummer sortiert werden
35      *              sollen.
36      */
37     private void sortiereRekursiv(int index, int anzahl) {
38         if (index == anzahl)
39             return;
40         int k = gibMinIndex(index, anzahl - 1);
41         if (k != index) {
42             vertausche(k, index);
43         }
44         sortiereRekursiv(index + 1, anzahl);
45     }
46
47     /**
48      * Sortiere ein Zahlen-Feld mit Hilfe des Selectionsort-Algorithmus in einer
49      * rekursiven Art und Weise.

```

```
48      *
49      * @return Das sortierte Zahlenfeld.
50      */
51      public int[] sortiere() {
52          sortiereRekursiv(0, zahlen.length);
53          return zahlen;
54      }
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/sortier/SelectionRekursiv.java](https://github.com/bschlangaul/sortier/SelectionRekursiv.java)

## Literatur

- [1] *Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19.* [https://www.studon.fau.de/file2567217\\_download.html](https://www.studon.fau.de/file2567217_download.html). FAU: Lehrstuhl für Informatik 2 (Programmiersysteme).
- [2] Gunter Saake und Kai-Uwe Sattler. *Algorithmen und Datenstrukturen. Eine Einführung in Java.* 2014.
- [3] Uwe Schneider. *Taschenbuch der Informatik.* 7. Aufl. Hanser, 2012. ISBN: 9783446426382.
- [4] *Wikipedia-Artikel „Selectionsort“.* <https://de.wikipedia.org/wiki/Selectionsort>.