

InsertionSort: Sortieren durch Einfügen

Weiterführende Literatur:

- Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 41
- Wikipedia-Artikel „Insertionsort“
- Saake und Sattler, Algorithmen und Datenstrukturen, Seite 125-127 (PDF 143-145)

Bei dem InsertionSort-Verfahren muss ein Element gemerkt werden. Dadurch wird eine Position in der Folge frei, die genutzt werden kann, um alle Elemente, die größer als das gemerkte Element sind, eine Position nach rechts zu verschieben. Das Verschieben erfolgt ausgehend von der aktuellen Position j rückwärts bis zum ersten Element. Ist das Element $j - 1$ kleiner oder gleich dem gemerkten Element, wird die innere Schleife verlassen. Durch das Verschieben „nach rechts“ wird die Position j in der Folge frei, an der das gemerkte Element eingefügt werden muss.¹

- Funktionsweise:
 - solange zu sortierende Liste mehr als ein Element beinhaltet:
 - lösche das *erste Element* aus der Liste
 - füge *gemäß Sortierordnung* in die Ergebnisliste ein
 - wiederhole, bis Eingangsliste leer
- Eigenschaften von InsertionSort:
 - Laufzeitkomplexität:
 - $\mathcal{O}(n)$ (im Best-Case)
 - $\mathcal{O}(n^2)$ (im Average- und Worst-Case)
 - *stabil*
 - bei Arrays *in-situ*

Minimal zum Auswendig lernen

```
11 public int[] sortiere() {
12     for (int i = 1; i < a.length; i++) {
13         int m = a[i];
14         int j = i;
15         while (j >= 1 && a[j - 1] > m) {
16             a[j] = a[j - 1];
17             j--;
18         }
19         a[j] = m;
20     }
21     return a;
22 }
```

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

¹Saake und Sattler, Algorithmen und Datenstrukturen, Seite 125.

Iterativ

```
9      /**
10     * Sortiere ein Zahlen-Feld mit Hilfe des Insertionsort-Algorithmus.
11     *
12     * @return Das sortierte Zahlenfeld.
13     */
14     public int[] sortiere() {
15         for (int i = 1; i < zahlen.length; i++) {
16             // Links von der Markierung sind die Zahlen sortiert,
17             // rechts davon unsortiert.
18             int markierung = zahlen[i];
19             int j = i;
20             // Für alle Zahlen links von der Markierung.
21             while (j >= 1 && zahlen[j - 1] > markierung) {
22                 // Die Zahl eins weiter nach rechts setzen
23                 // An der Position j - 1 und j stehen jetzt zweimal die gleichen
24                 // Zahlen
25                 zahlen[j] = zahlen[j - 1];
26                 j--;
27             }
28             // Die markierte Zahl an die richtige Stelle setzen.
29             zahlen[j] = markierung;
30         }
31         return zahlen;
32     }
```

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

2

Zustand des Eingabefelds an einem Beispiel

```
1  for: i=1
2  for (Anfang) 7 4 9 2 3
3  while        7 7 9 2 3
4  for (Ende)   4 7 9 2 3
5  for: i=2
6  for (Anfang) 4 7 9 2 3
7  for (Ende)   4 7 9 2 3
8  for: i=3
9  for (Anfang) 4 7 9 2 3
10 while        4 7 9 9 3
11 while        4 7 7 9 3
12 while        4 4 7 9 3
13 for (Ende)   2 4 7 9 3
14 for: i=4
15 for (Anfang) 2 4 7 9 3
16 while        2 4 7 9 9
17 while        2 4 7 7 9
18 while        2 4 4 7 9
19 for (Ende)   2 3 4 7 9
```

Rekursiver Insertionsort

```
10     /**
11     * @param i Wie viele Zahlen im Feld sortiert werden sollen.
```

²Saake und Sattler, *Algorithmen und Datenstrukturen*, Seite 125 - 127.

```

12  */
13  private void sortiereRekursiv(int i) {
14      // Abbruchsfall
15      if (i <= 1)
16          return;
17
18      // Sortiere die ersten n-1 Elemente.
19      sortiereRekursiv(i - 1);
20
21      int markierung = zahlen[i - 1];
22      int j = i - 2;
23
24      while (j >= 0 && zahlen[j] > markierung) {
25          zahlen[j + 1] = zahlen[j];
26          j--;
27      }
28      zahlen[j + 1] = markierung;
29  }
30
31  /**
32   * Sortiere ein Zahlen-Feld mit Hilfe des Insertionsort-Algorithmus in einer
33   * rekursiven Art und Weise.
34   *
35   * @return Das sortierte Zahlenfeld.
36   */
37  public int[] sortiere() {
38      sortiereRekursiv(zahlen.length);
39      return zahlen;
40  }

```

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

Literatur

- [1] *Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19.* 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] *Wikipedia-Artikel „Insertionsort“.* <https://de.wikipedia.org/wiki/Insertionsort>.