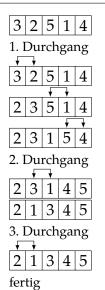
# **BubbleSort: Blasensortierung**

#### Weiterführende Literatur:

- Algorithmen und Datenstrukturen: Tafelübung 11, WS 2018/19, Seite 43
- Saake und Sattler, Algorithmen und Datenstrukturen, Seite 129-131 (PDF 147-149)
- Wikipedia-Artikel "Bubblesort"



### **Funktionsweise**

1 2 3 4 5

- solange zu sortierende Liste nicht vollständig sortiert ist:
- iteriere von vorne über die Elemente
- falls zwei aufeinanderfolgende Elemente nicht sortiert sind: vertauschen
- fertig, wenn während einer Runde keine Vertauschung passiert ist

## Eigenschaften

- Laufzeitkomplexität:
  - $\mathcal{O}(n)$  (im Best-Case)
  - $\mathcal{O}(n^2)$  (im Average- und Worst-Case)
- stabil
- in-situ

```
10
       * Sortiere mit Hilfe des Bubblesort-Algorithmus.
11
12
       * <strong>Abkürzungen:</strong>
13
14
15
       * t: getauscht
16
17
       * 
18
      public int[] sortiere() {
19
        boolean t;
        do {
21
22
          t = false;
23
          for (int i = 0; i < a.length - 1; i++) {
            if (a[i] > a[i + 1]) {
24
25
               vertausche(i, i + 1);
26
               t = true;
            }
27
          }
28
        } while (t);
29
30
        return a;
31
                                  Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/sortier/BubbleMinimal.java
       * Sortiere ein Zahlen-Feld mit Hilfe des Bubblesort-Algorithmus.
10
11
       * Oreturn Das sortierte Zahlenfeld.
12
13
14
      public int[] sortiere() {
        boolean getauscht;
15
16
        do {
17
          getauscht = false;
          for (int i = 0; i < zahlen.length - 1; i++) {
18
19
            if (zahlen[i] > zahlen[i + 1]) {
20
               // Elemente vertauschen
               vertausche(i, i + 1);
21
               getauscht = true;
23
24
25
          // solange Vertauschung auftritt
        } while (getauscht);
26
27
        return zahlen;
                                  Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/sortier/BubbleIterativ.java
    1
11
      private void sortiereRekursiv(int[] zahlen, int letztes) {
12
        // Bei einem leeren Feld kann die Variable letztes -1 annehmen.
13
14
        // Wenn letztes 0 ist, dann sind die Zahlen sortiert.
        if (letztes <= 0)</pre>
15
          return;
16
17
        // Ein Durchgang der Blasensortierung. Nach diesem Durchgang befindet sich
18
        // größte Element am Ende des Felds.
        for (int i = 0; i < letztes; i++)</pre>
20
```

<sup>&</sup>lt;sup>1</sup>Saake und Sattler, Algorithmen und Datenstrukturen, Seite 130-131.

```
if (zahlen[i] > zahlen[i + 1]) {
    vertausche(i, i + 1);
}

// Sortiere jetzt die restlichen Elemente.
sortiereRekursiv(zahlen, letztes - 1);

public int[] sortiere() {
    sortiereRekursiv(zahlen, zahlen.length - 1);
    return zahlen;
```

Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/sortier/BubbleRekursiv.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 "Bubblesort".https://de.wikipedia.org/wiki/Bubblesort.