

Aufgabe 4

Bei Bubblesort wird eine unsortierte Folge von Elementen a_1, a_2, \dots, a_n , von links nach rechts durchlaufen, wobei zwei benachbarte Elemente a_i und a_{i+1} getauscht werden, falls sie nicht in der richtigen Reihenfolge stehen. Dies wird so lange wiederholt, bis die Folge sortiert ist.

- (a) Sortieren Sie die folgende Zahlenfolge mit Bubblesort. Geben Sie die neue Zahlenfolge nach jedem (Tausch-)Schritt an: 3, 2, 4, 1

```
3 2 4 1 Eingabe
3 2 4 1 Durchlauf Nr. 1
>3 2< 4 1 vertausche (i 0<>1)
2 3 >4 1< vertausche (i 2<>3)
2 3 1 4 Durchlauf Nr. 2
2 >3 1< 4 vertausche (i 1<>2)
2 1 3 4 Durchlauf Nr. 3
>2 1< 3 4 vertausche (i 0<>1)
1 2 3 4 Durchlauf Nr. 4
1 2 3 4 Ausgabe
```

- (b) Geben Sie den Bubblesort-Algorithmus für ein Array von natürlichen Zahlen in einer Programmiersprache Ihrer Wahl an. Die Funktion `swap (index1, index2)` kann verwendet werden, um zwei Elemente des Arrays zu vertauschen.

```
3 public class BubbleSort {
4
5     public static void swap(int[] array, int index1, int index2) {
6         int tmp = array[index1];
7         array[index1] = array[index2];
8         array[index2] = tmp;
9     }
10
11     public static void bubblesort(int[] array) {
12         boolean swapped;
13         do {
14             swapped = false;
15             for (int i = 0; i < array.length - 1; i++) {
16                 if (array[i] > array[i + 1]) {
17                     swap(array, i, i + 1);
18                     swapped = true;
19                 }
20             }
21         } while (swapped);
22     }
23 }
```

Code-Beispiel auf Github ansehen:
[src/main/java/org/bschlangaul/examen/examen_46115/jahr_2017/fruehjahr/BubbleSort.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2017/fruehjahr/BubbleSort.java)

Test

```

3  import static org.junit.Assert.assertEquals;
4
5  import org.junit.Test;
6
7  public class BubbleSortTest {
8
9      @Test
10     public void teste() {
11         int[] array = new int[] { 3, 2, 4, 1 };
12         BubbleSort.bubblesort(array);
13         assertEquals(1, array[0]);
14         assertEquals(2, array[1]);
15         assertEquals(3, array[2]);
16         assertEquals(4, array[3]);
17     }
18 }

```

Code-Beispiel auf Github ansehen:
[src/test/java/org/bschlangaul/examen/examen_46115/jahr_2017/fruehjahr/BubbleSortTest.java](https://github.com/bschlangaul/examen_exam_46115_jahr_2017_fruehjahr/BubbleSortTest.java)

- (c) Geben Sie eine obere Schranke für die Laufzeit an. Beschreiben Sie mögliche Eingabedaten, mit denen diese Schranke erreicht wird.

$$\mathcal{O}(n^2)$$

Diese obere Schranke wird erreicht, wenn die Zahlenfolgen in der umgekehrten Reihenfolge bereits sortiert ist, z. B. 4, 3, 2, 1.