

## Aufgabe 8

Gegeben sei das folgende Feld  $A$  mit 7 Schlüsseln:

[15, 4, 10, 7, 1, 8, 10]

- (a) Sortieren Sie das Feld mittels des Sortierverfahrens *Bubblesort*. Markieren Sie jeweils, welche zwei Feldwerte verglichen werden und geben Sie den Zustand des gesamten Feldes jeweils neu an, wenn Sie eine Vertauschung durchgeführt haben.
- (b) Sortieren Sie das Feld mittels des Sortierverfahrens *Selectionsort*. Markieren Sie jeweils, welche zwei Feldwerte verglichen werden und geben Sie den Zustand des gesamten Feldes jeweils neu an, wenn Sie eine Vertauschung durchgeführt haben.
- (c) Vergleichen Sie beide Sortierverfahren hinsichtlich ihres Laufzeitverhaltens im *best case*. Welches Verfahren ist in dieser Hinsicht besser, wenn das zu sortierende Feld anfangs bereits sortiert ist? Begründen Sie Ihre Antwort.

Der Bubblesort-Algorithmus hat im *best case* eine Laufzeit von  $\mathcal{O}(n)$ , der Selectionsort-Algorithmus  $\mathcal{O}(n^2)$ .

Bubblesort steuert seine äußere bedingte Wiederholung in vielen Implementationen über eine boolsche Hilfsvariable `getauscht`, die beim Betreten der Schleife erstmals auf falsch gesetzt wird. Erst wenn Vertauschungen vorgenommen werden müssen, wird diese Variable auf wahr gesetzt und die äußere Schleife läuft ein weiteres Mal ab. Ist das zu sortierende Feld bereits sortiert, durchsucht der Algorithmus das Feld einmal und terminiert dann.

Der Selectionsort-Algorithmus hingegen ist mit zwei ineinander verschränkten Schleifen umgesetzt, deren Wiederholungsanzahl sich starr nach der Anzahl der Elemente im Feld richtet.

### Bubblesort

```
15     boolean getauscht;  
16     do {  
17         getauscht = false;  
18         for (int i = 0; i < zahlen.length - 1; i++) {  
19             if (zahlen[i] > zahlen[i + 1]) {  
20                 // Elemente vertauschen  
21                 vertausche(i, i + 1);  
22                 getauscht = true;  
23             }  
24         }  
25         berichte.zahlenFeld();  
26         // solange Vertauschung auftritt  
27     } while (getauscht);
```

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

## Selectionsort

```
12 // Am Anfang ist die Markierung das letzte Element im
13 ↪ Zahlen-Array.
14 int markierung = zahlen.length - 1;
15 while (markierung >= 0) {
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     // Tausche zahlen[markierung] mit dem gefundenem Element.
27     vertausche(markierung, max);
28     // Die Markierung um eins nach vorne verlegen.
29     markierung--;
30 }
```

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