

66115 / 2018 / Frühjahr

Thema 2 / Aufgabe 8 (Sortieren von 15,4,10,7,1,8,10 mit Bubble- und Selectionsort)**Stichwörter:** Sortieralgorithmen, Bubblesort, Selectionsort, Algorithmische Komplexität (O-Notation)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.

Lösungsvorschlag

15	4	10	7	1	8	10	Eingabe
15	4	10	7	1	8	10	Durchlauf Nr. 1
>15	4<	10	7	1	8	10	vertausche (i 0<>1)
4	>15	10<	7	1	8	10	vertausche (i 1<>2)
4	10	>15	7<	1	8	10	vertausche (i 2<>3)
4	10	7	>15	1<	8	10	vertausche (i 3<>4)
4	10	7	1	>15	8<	10	vertausche (i 4<>5)
4	10	7	1	8	>15	10<	vertausche (i 5<>6)
4	10	7	1	8	10	15	Durchlauf Nr. 2
4	>10	7<	1	8	10	15	vertausche (i 1<>2)
4	7	>10	1<	8	10	15	vertausche (i 2<>3)
4	7	1	>10	8<	10	15	vertausche (i 3<>4)
4	7	1	8	10	10	15	Durchlauf Nr. 3
4	>7	1<	8	10	10	15	vertausche (i 1<>2)
4	1	7	8	10	10	15	Durchlauf Nr. 4
>4	1<	7	8	10	10	15	vertausche (i 0<>1)
1	4	7	8	10	10	15	Durchlauf Nr. 5
1	4	7	8	10	10	15	Ausgabe

- (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.

Lösungsvorschlag

15	4	10	7	1	8	10	Eingabe
15	4	10	7	1	8	10*	markiere (i 6)
>15	4	10	7	1	8	10<	vertausche (i 0<>6)
10	4	10	7	1	8*	15	markiere (i 5)
>10	4	10	7	1	8<	15	vertausche (i 0<>5)
8	4	10	7	1*	10	15	markiere (i 4)
8	4	>10	7	1<	10	15	vertausche (i 2<>4)
8	4	1	7*	10	10	15	markiere (i 3)
>8	4	1	7<	10	10	15	vertausche (i 0<>3)
7	4	1*	8	10	10	15	markiere (i 2)

>7	4	1<	8	10	10	15	vertausche (i 0<>2)
1	4*	7	8	10	10	15	markiere (i 1)
1	>4	7	8	10	10	15	vertausche (i 1<>1)
1*	4	7	8	10	10	15	markiere (i 0)
>1	4	7	8	10	10	15	vertausche (i 0<>0)
1	4	7	8	10	10	15	Ausgabe

- (c) Vergleichen Sie beide Sortiervverfahren 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

```
int durchlaufNr = 0;
boolean getauscht;
do {
    durchlaufNr++;
    berichte.feld("Durchlauf Nr. " + durchlaufNr);
    getauscht = false;
    for (int i = 0; i < zahlen.length - 1; i++) {
        if (zahlen[i] > zahlen[i + 1]) {
            // Elemente vertauschen
            vertausche(i, i + 1);
            getauscht = true;
        }
    }
}
```

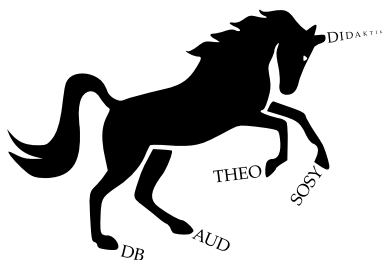
Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/sortier/BubbleIterativ.java](https://github.com/bschlangaul/sortier/BubbleIterativ.java)

Selectionsort

```
// Am Anfang ist die Markierung das letzte Element im Zahlen-Array.
int markierung = zahlen.length - 1;
while (markierung >= 0) {
    berichte.feldMarkierung(markierung);
}
```

```
// Bestimme das größtes Element.  
// max ist der Index des größten Elements.  
int max = 0;  
// Wir vergleichen zuerst die Zahlen mit der Index-Number  
// 0 und 1, dann 1 und 2, etc. bis zur Markierung  
for (int i = 1; i <= markierung; i++) {  
    if (zahlen[i] > zahlen[max]) {  
        max = i;  
    }  
}  
  
// Tausche zahlen[markierung] mit dem gefundenem Element.  
vertausche(markierung, max);  
// Die Markierung um eins nach vorne verlegen.  
markierung--;
```

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



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht alleine! Das ist ein Community-Projekt. Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der \LaTeX -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/hbschlang/lehramt-informatik/blob/main/Staatsexamen/66115/2018/09/Thema-2/Aufgabe-8.tex>