

# Algorithmen und Datenstrukturen

## Sortieren 1

Die drei einfachen Sortierverfahren Bubble Sort, Selection Sort und Insertion Sort sind alle für ungeordnete Datenbestände von der Ordnung  $O(n^2)$ . Dennoch unterscheiden sie sich in ihren Laufzeiten, was sich aus der Theorie ableiten lässt. Sie sollen in diesem Praktikum diese Unterschiede durch Laufzeit-Messungen quantifizieren.

### Aufgabe 1 Bubble Sort mit Testdaten [2 Punkte]

Schreiben Sie eine Klasse `SortServer`, die den `CommandExecutor` implementiert. Es sollen Datenbestände unterschiedlicher Grösse sortiert werden können, deren Schlüssel durch einen Array von Ganzzahlen mit Werten im Bereich  $[0..1'000'000[$  repräsentiert werden. Implementieren Sie den Bubble Sort aus dem Skript. Die Anzahl der zu sortierenden Daten soll als Parameter mitgegeben werden können. Schreiben Sie eine Methode `checkSorted`, die überprüft, ob das Array auch korrekt sortiert wurde. Das Resultat der Überprüfung soll das Resultat der `execute` Methode sein.

#### Hinweis:

- Die Methode `nextInt(<Bereich>)` der Klasse `Random` kann zur Erzeugung von Zufallszahlen verwendet werden.

### Aufgabe 2 Messen der Laufzeit [3 Punkte]

Die Laufzeit des Sortiervorgangs soll in Millisekunden gemessen werden.

#### Hinweise:

- Um die Laufzeit zu bestimmen, kann die Methode `currentTimeMillis` der Klasse `System` verwendet werden. Da jedoch lediglich in einer Auflösung von Millisekunden gemessen werden kann, muss für kleine Datenbestände der Aufruf der Sortiermethode mehrmals wiederholt werden. Da wir über einen grossen Bereich messen wollen, ist es praktischer, nicht die Anzahl Wiederholungen sondern die Messzeit konstant zu halten. Ein weiteres Problem der Zeitmessung kann man folgendem Auszug aus der Java Dokumentation zu obiger Methode entnehmen: "*The granularity of the value depends on the underlying operating system and may be larger. For example, many operating systems measure time in units of tens of milliseconds.*"
- Verwenden Sie folgendes Gerüst für Ihre Messroutine:

```
long end, start = System.currentTimeMillis()
int count = 0;
do {
    <Operation>
    count++;
    end = System.currentTimeMillis();
} while (end - start < 1000);
System.out.println("time="+((double)(end-start)/count));
```

- Damit die Erstellung des Datenbestandes nicht die Messung verfälscht, muss dieser einmal erzeugt und anschliessend jeweils eine Kopie sortiert werden. Mittels der Methode `arraycopy` der Klasse `System` lässt sich ein Array schnell kopieren.

### **Aufgabe 3 Selection Sort und Insertion Sort [3 Punkte]**

Es sollen die beiden anderen Sortieralgorithmen ebenfalls implementiert werden. Der `execute` Methode soll angegeben werden können, welcher Algorithmus angewandt werden soll, z.B.

```
INSERTION 10000  
BUBBLE 10000
```

### **Aufgabe 4 Darstellung in Excel [2 Punkte]**

Stellen Sie die erhaltenen Daten graphisch dar. Sie können die Resultate für diesen Zweck ins Excel übernehmen. Wählen Sie einen geeigneten Diagrammtyp.