

# Algorithmen und Datenstrukturen INF3/ICS3

## Wintersemester 2023/24

Prof. Dr. Georg Schied

## Aufgabenblatt 5

**Abgabetermin: Mo. 20. November 2023, 23:59 Uhr**  
Zum Bestehen müssen 8 von 16 Punkten erreicht werden.

### Aufgabe 5.1

- a) Ein Array mit den Werten **5, 9, 4, 7, 1, 6, 8** ist gegeben. Wie würden diese Daten mit der Methode `aufteilen()` in zwei Bereiche aufgeteilt werden? Geben Sie den Arrayinhalt nach jeder Iteration der Schleife an.
- b) Stellen Sie dar, wie die Werte mittels Quicksort sortiert werden. Das Aufteilen muss nicht genau nach Methode `aufteilen()` gemacht werden, es sollte aber erkennbar sein, wie jeweils das Pivot-Element gewählt wird.

### Aufgabe 5.2 - Scheinaufgabe (5 Punkte)

Gegeben ist folgendes Array:

2	9	3	1	4	8	5	7	6
---	---	---	---	---	---	---	---	---

Stellen Sie dar, wie die Daten mit **Quicksort** sortiert werden. Das Aufteilen muss *nicht exakt* nach der in der Vorlesung vorgestellten Methode `aufteilen()` vorgenommen werden. Es soll aber erkennbar sein, nach welcher Regel das Pivotelement gewählt wird.

### Aufgabe 5.3

Gegeben ist ein Array mit folgenden Werten:

**7, 6, 5, 4, 3, 2, 1**

Wie würden die Werte mittels **Mergesort** sortiert werden?

### Aufgabe 5.4 - Scheinaufgabe (5 Punkte)

Gegeben ist folgendes Array:

2	9	3	1	4	8	5	7	6
---	---	---	---	---	---	---	---	---

Stellen Sie dar, wie die Daten mit **Mergesort** sortiert werden. Geben Sie jeweils die wesentlichen Zwischenschritte an, so dass das Teile-und-Herrsche-Prinzip erkennbar ist.

## Aufgabe 5.5

---

[weiterführend, nicht so einfach] Der Median einer Menge von Werten ist der  $\lceil n/2 \rceil$ -kleinste Wert, d.h. der Wert in der Mitte, wenn die Daten in sortierter Reihenfolge wären. Wie kann der Median effizient berechnet werden? Gibt es ein  $O(n)$ -Verfahren dafür?

- a) Entwerfen Sie eine effiziente Methode zur Berechnung des Medians eines Felds `a`:

```
public static double median(double[] a)
```

- b) Analysieren Sie, in welcher Größenordnung die Laufzeit der Methode liegt.
- c) Messen Sie die Laufzeit für Felder der Größe  $n = 100, 1000, \dots, 1\,000\,000$ . Verwenden Sie dazu Felder mit zufällig gefüllten Daten. Vergleichen Sie die gemessenen Werte mit der theoretisch analysierten Größenordnung für die Laufzeit.

In Moodle finden Sie eine Programmvorlage zur Laufzeitmessung und ein JUnit-Testprogramm.

## Aufgabe 5.6 - Scheinaufgabe (6 P)

---

Messen Sie die Geschwindigkeit verschiedener vergleichsbasierter Sortierverfahren. In Moodle finden Sie die Klasse `Sort`, die Implementierungen für Selectionsort, Insertionsort, Bubblesort, Heapsort, Quicksort und Mergesort enthält sowie die Klasse `SortMeasurement`, mit der die Laufzeit für das Sortieren bei verschiedenen Problemgrößen gemessen werden kann.

- a) Bestimmen Sie für die verschiedenen Sortierverfahren die Laufzeit für das Sortieren von Feldern mit  $n = 100, 1000, 10\,000$  und  $100\,000$  Werten. Messen Sie die Laufzeit bei Ausführung mit Option `-Xint`. Geben Sie eine Tabelle mit Ihren Messergebnissen ab und erläutern Sie kurz, ob das erwartete Laufzeitverhalten erkennbar ist.
- b) Schätzen Sie, ausgehend von den gemessenen Werten, für
- (1) Insertionsort
  - (2) Bubblesort
  - (3) Heapsort

die Laufzeiten für das Sortieren von  $n = 1\,000\,000$  und  $n = 10\,000\,000$  Einträgen. (mit kurzer Erläuterung). Geben Sie die Zeiten, falls nicht im Millisekundenbereich, in passenden anderen Einheiten (Sekunden, Minuten, Stunden, ...) an.

## Aufgabe 5.7

---

Zeigen Sie, wie die Werte

7, 6, 40, 13, 77, 9, 55, 25, 33, 14, 8, 66, 13, 85, 22, 15, 3, 64, 32

mit ausgeglichenem 2-Wege-Mergesort sortiert werden. Dazu sei angenommen, dass  $k=4$  Werte auf einmal im Hauptspeicher sortiert werden können.