

Aufgabe 6.1

In der Vorlesung sind wir bei der Implementierung des Algorithmus MergeSort sehr verschwenderisch mit dem Speicherplatz umgegangen. Versuchen Sie den Speicherplatzbedarf des Algorithmus zu verringern, indem Sie direkt auf dem Eingabefeld arbeiten. Dazu sollen Sie Methoden

```
static void MergeSort(int[] f, int u, int o)
```

und

```
static void MischenArrays(int[] f, int u1, int o1, int u2, int o2,)
```

implementieren, die im Wesentlichen den entsprechenden Methoden aus der Vorlesung folgen, aber eben mit den Parametern u und o , bzw. $u1$, $o1$, $u2$ und $o2$ den Bereich bzw. die Bereiche des Arrays f markieren, die sortiert bzw. gemischt werden sollen. Speichern Sie die neuen Methoden in einer Klasse `MischMasch.java`, die Sie in Analogie zum Beispiel aus der Vorlesung gestalten können. Arbeiten Sie Ihre Variante des MergeSort Algorithmus *in-place*? Wo ergaben sich ggfs. Schwierigkeiten, dies zu erreichen?

(8 Punkte)

Aufgabe 6.2

Zeigen Sie für die folgenden beiden Funktionen f und g , dass $f \in O(g)$ ist.

$$f(n) = 10n^2 + 3n + 6$$

$$g(n) = n^3 - 9$$

(3 Punkte)

Aufgabe 6.3

Bringen Sie die folgenden sieben Funktionen in eine Reihenfolge f_1, f_2, \dots, f_7 , sodass $f_i \in O(f_{i+1})$ für alle $i \in \{1, 2, \dots, 6\}$ gilt.

$$n^2, \quad \log_2 n, \quad 2^n, \quad \sqrt{n}, \quad n^n, \quad \sqrt[3]{n}, \quad n \log_2 n$$

(3 Punkte)

Zusatzaufgabe 6.4

Im Material zu diesem Aufgabenblatt finden Sie das Programm `Fibonacci.java`. Dieses berechnet für die als Kommandozeilenparameter übergebene natürliche Zahl n die Fibonacci-Zahl F_n . Die Fibonacci-Zahlen sind folgendermaßen induktiv definiert:

$$F_n = \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ F_{n-1} + F_{n-2} & \text{falls } n \geq 2 \end{cases}$$

Das Programm `Fibonacci.java` berechnet F_n , indem diese Definition direkt in eine rekursive Methode

```
public static long fib(int n)
```

umgesetzt wird. Wenn man das Programm nun laufen lässt, stellt man fest, dass die Berechnung schon für relativ kleine n (zwischen 40 und 50 etwa) relativ lange dauert. Woran könnte das liegen?

(3+Punkte)