



## Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 7– 17.06.2019

Abgabe: Briefkästen RZ/E-Mail bis Montag, den 24.06.2019, 16:00 Uhr

---

**Aufgabe 1** (1+2+2 Punkte). Der Aufwand eines Algorithmus zur Lösung einer Aufgabe der Größe  $n$  ist definiert als die Anzahl  $\mathcal{A}(n)$  der erforderlichen arithmetischen Operationen und Vergleiche.

(i) Ordnen Sie die folgenden Schranken für den Aufwand eines Algorithmus in der Reihenfolge ihrer Günstigkeit für große Probleme der Größe  $n$ :

$$n^2, \quad 2^n, \quad n \log(n), \quad n^3, \quad n!, \quad n, \quad 1, \quad \log(n).$$

Begründen Sie Ihre Antwort!

(ii) Bestimmen Sie den Aufwand jeweils für die dynamische und (naive) rekursive Berechnung der  $n$ -ten Fibonacci-Zahl  $f_n$ .

(iii) Eine mögliches Maß für die Leistungsfähigkeit eines Computers ist die Einheit *Floating Point Operations Per Second* (kurz *FLOPS*; engl. für Gleitkommaoperationen pro Sekunde). Es gibt an, wie viele Gleitkommaoperationen (d. h. Additionen und Multiplikationen) ein Rechner pro Sekunde ausführen kann. Beispielsweise erreicht ein Pentium 4 Prozessor mit 3,2 GHz aus dem Jahr 2004 eine Höchstleistung von 6,4 GigaFLOPS, ein Core i7 4-Kern-Prozessor mit 3,4 GHz aus dem Jahr 2011 eine Höchstleistung von 92,3 GigaFLOPS, und der Cray XC40 Supercomputer „Hazel Hen“ am Höchstleistungsrechenzentrum Stuttgart, der aktuell der schnellste Rechner Deutschlands ist, eine Höchstleistung von 5,6 PetaFLOPS. Die Vorsilbe „Giga“ steht dabei für den Faktor  $10^9$  und die Vorsilbe „Peta“ für den Faktor  $10^{15}$ . Berechnen Sie auf dieser Grundlage die Mindestzeit, die für die rekursive Berechnung der 90. Fibonacci-Zahl  $f_{90}$  auf jedem dieser Systeme benötigt wird.

**Aufgabe 2** (5 Punkte). Betrachten Sie die beiden folgenden Sortieralgorithmen, welche jeweils eine ungeordnete Liste  $L = [a_1, a_2, \dots, a_n]$  mit  $n$  Einträgen in eine geordnete Reihenfolge bringen. Dabei können die Einträge der Liste beispielsweise Zahlen oder Wörter sein, die in eine aufsteigende bzw. alphabetische Reihenfolge gebracht werden sollen.

**Algorithmus** (Bubblesort). *Eingabe:* Liste  $L = [a_1, a_2, \dots, a_n]$ .

- (1) Setze  $i = 2$ .
- (2) Setze  $k = 0$ .
- (3) Falls  $a_{i-k} < a_{i-k-1}$ , so vertausche die beiden Einträge, andernfalls gehe zu (6).
- (4) Falls  $k < i - 2$ , so erhöhe  $k$  um 1 und gehe zu (3).
- (5) Falls  $i < n$ , so erhöhe  $i$  um 1 und gehe zu (2).

*Ausgabe:* Sortierte Liste  $L = [a_1, a_2, \dots, a_n]$ .

---

Fortsetzung auf der Rückseite.

**Algorithmus** (Mergesort). *Eingabe:* Liste  $L$  der Länge  $n = 2^k$ , wobei  $k \geq 0$ .  
 Aufruf des Algorithmus durch  $\text{msort}(L, k)$ .

- (1) Gilt  $k = 0$ , so ist die Liste bereits sortiert, das heißt setze  $\tilde{L} = L$ .
- (2) Gilt  $k > 0$ , so setze

$$\tilde{L} = \text{merge}(\text{msort}(L_{n/2}^{\ell}, k - 1), \text{msort}(L_{n/2}^r, k - 1)).$$

*Ausgabe:* Sortierte Liste  $\tilde{L}$ .

Dabei bezeichnen  $L^{\ell} = [a_1, a_2, \dots, a_{2^{r-1}}]$  und  $L^r = [a_{2^{r-1}+1}, \dots, a_{2^r}]$  jeweils die linke bzw. rechte Hälfte einer Liste  $L$  und der Befehl  $\text{merge}(A, B)$  steht für das Zusammenfügen zweier bereits sortierter Listen  $A, B$  zu einer einzigen sortierten Liste.

Erläutern Sie die Sortieralgorithmen Bubblesort und Mergesort am Beispiel der folgenden Liste von Namen, welche in eine alphabetische Reihenfolge gebracht werden soll (die mathematischen Vergleiche „<“ und „>“ sind in diesem Zusammenhang als lexikographische Vergleiche zu deuten):

[ Hannes, Marie, Jan, Lukas, Josepha, Emanuel, Tatjana, David ]

Sortieren Sie dazu die Liste mit beiden Algorithmen per Hand und schreiben Sie nach jedem Schritt jeweils die aktuelle Reihenfolge der Elemente sowie die Operationen für den darauffolgenden Schritt auf.

**Aufgabe 3** (3+2 Punkte). Die Datei `temperaturen.txt` auf der Vorlesungshomepage enthält die Jahresdurchschnittstemperaturen in Deutschland in Grad Celsius aus den Jahren 1960-2018.

(i) Schreiben Sie ein Programm, welches diese Datei einliest und in zwei Vektoren oder Arrays speichert und berechnet Sie die durchschnittliche Jahrestemperatur.

(ii) Erweitern Sie ihr Programm, sodass alle Jahre samt Temperaturen in eine neue Datei, z. B. `auffaellig.txt`, schreibt, sofern die Abweichung von der Durchschnittstemperatur höher als  $0.75^\circ \text{C}$  ist.

**Aufgabe 4** (4+1 Punkte). (i) Implementieren Sie den Bubblesort-Algorithmus aus Aufgabe 2, um einen Vektor oder ein Array beliebiger Länge zu sortieren. Testen Sie Ihr Programm mit dem Vektor der Jahresdurchschnittstemperaturen aus Aufgabe 3.

(ii) Erweitern Sie das Programm aus Aufgabe 3 um den Bubblesort-Algorithmus und schreiben Sie die sortierten Temperaturen mit den zugehörigen Jahreszahlen in eine neue Datei. (Hinweis: Hierbei müssen die jeweiligen Jahreszahlen im Bubblesort-Algorithmus entsprechend den Temperaturen vertauscht werden.)