

Übungsblatt Nr. 2

Abgabe der Lösungen in Moodle bis zum 26.10.2015 um 19 Uhr (+n). Die Lösungen werden am 27.10.2015 in der Übung vorgestellt.

Aufgabe 1

Die Primzahlen in einem Intervall ganzer Zahlen $\{2, \dots, n\}$ lassen sich so berechnen, dass zunächst alle Zahlen aus dem Intervall entfernt werden, die durch 2 teilbar sind und größer als 2 sind. Aus der Menge der verbleibenden Zahlen werden dann alle diejenigen Zahlen entfernt, die durch 3 teilbar sind und größer als 3 sind, usw., bis $n/2$ erreicht ist.

- Schreiben Sie einen Algorithmus in Pseudocode, der dies leistet.
- Überlegen Sie sich, wie lange der Algorithmus in Abhängigkeit von n maximal dauern könnte. Erweitern Sie Ihren Pseudocode um einen Zähler.
- Legen Sie eine Tabelle an, in der Sie angeben, wieviele Schritte Sie für $n=20$, $n=50$, $n=70$ und $n=100$ benötigen.
- Haben Sie diesen Algorithmus aus der Mathematik erkannt? Wie heißt er?

Aufgabe 2

Ein bedeutsamer Sortieralgorithmus ist *MergeSort*. Eine mögliche Pseudocode-Implementierung von MergeSort ist (siehe <https://de.wikipedia.org/wiki/Mergesort>):

```
funktion mergesort(liste);
  falls (Größe von liste <= 1) dann antworte liste
  sonst
    halbiere die liste in linkeListe, rechteListe
    linkeListe = mergesort(linkeListe)
    rechteListe = mergesort(rechteListe)
    antworte merge(linkeListe, rechteListe)

funktion merge(linkeListe, rechteListe);
  neueListe
  solange (linkeListe und rechteListe nicht leer)
  |   falls (erstes Element der linkeListe <= erstes Element der rechteListe)
  |   dann füge erstes Element linkeListe in die neueListe hinten ein und entferne es aus linkeListe
  |   sonst füge erstes Element rechteListe in die neueListe hinten ein und entferne es aus rechteListe
  solange_ende
  solange (linkeListe nicht leer)
  |   füge erstes Element linkeListe in die neueListe hinten ein und entferne es aus linkeListe
  solange_ende
  solange (rechteListe nicht leer)
  |   füge erstes Element rechteListe in die neueListe hinten ein und entferne es aus rechteListe
  solange_ende
  antworte neueListe
```

Angenommen wir haben $n = 10$ natürliche Zahlen in beliebiger Reihenfolge gegeben (z.B. (5, 9, 2, 10, 3, 1, 4, 7, 6, 8)) und wollen diese sortieren. Wir lesen sie in ihrer Reihenfolge als Liste in den obigen Algorithmus ein.

- Wie viele Vergleiche sind mit dieser Implementierung notwendig, um die Zahlen zu sortieren?
- Verallgemeinern Sie das Resultat für n natürliche Zahlen.

Aufgabe 3

- a) Die Summe der ganzen Zahlen zwischen 1 und n soll berechnet werden. Schreiben Sie einen Algorithmus in Pseudocode, der dies umsetzt.
- b) Die Summe S_n der ganzen Zahlen zwischen 1 und n kann rekursiv wie folgt definiert werden:

$$S_1 := 1,$$
$$S_n := S_{n-1} + n \text{ für } n > 1$$

Schreiben Sie einen Algorithmus in Pseudocode, der die Berechnung auf diese Weise ausführt.

Aufgabe 4

Algorithmen können auch graphisch mit Programmablaufplänen (Flowcharts) dargestellt werden.

- a) Testen Sie die folgenden Eingaben: $n=140$ und $m=63$ mit dem untenstehenden Programm und schreiben Sie die Zwischenwerte der Variablen r und die sich verändernden Werte von m und n in einer Tabelle auf. Was ist am Ende das Ergebnis?
- b) Welchen bekannten Algorithmus repräsentiert dieser Ablaufplan?

