

## Übungsblatt 2

Abgabe bis Dienstag, den 6. Mai 2025 um 12:00 Uhr

*Wichtige Vorbemerkung: Sie lernen nichts, wenn Sie sich die Lösung für die folgenden Aufgaben irgendwo anschauen oder erklären lassen. Probieren Sie selber darauf zu kommen und die Beweise selber zu führen. Es sind sehr schöne Aufgaben und eine schöne Gelegenheit, diese Dinge zu lernen und einzuüben. Wenn Sie nach einer angemessenen Zeit gar nicht weiterkommen, lassen Sie sich einen kleinen Tipp geben, aber nicht die komplette Lösung. Sie lernen nur etwas, wenn Sie es selber probieren und auch länger selber probieren. Wenn Ihnen diese Aufgaben alle viel zu schwer sind, tut uns das leid, aber dann fehlen Ihnen wahrscheinlich wesentliche Grundlagen, wie zu Beginn der ersten Vorlesung besprochen.*

### Aufgabe 1 (5 Punkte)

In der Vorlesung haben wir gezeigt, dass es eine Konstante  $C > 0$  gibt, so dass die Laufzeit von *MinSort* für Eingabegröße  $n$  immer  $\leq C \cdot n^2$  ist. Zeigen Sie, dass auch gilt, dass es eine Konstante  $C' > 0$  gibt, so dass die Laufzeit von *MinSort* für Eingabegrößen  $n \geq 2$  immer  $\geq C' \cdot n^2$  ist.

### Aufgabe 2 (5 Punkte)

Sei  $b(n)$  die Anzahl Stellen (Bits) der Binärdarstellung von  $n \in \mathbb{N}$ ,  $n > 0$ . Zum Beispiel ist  $b(7) = 3$  und  $b(8) = 4$ . Finden Sie eine geschlossene Formel für  $b(n)$  in Abhängigkeit von  $n$  und beweisen Sie, dass sie korrekt ist.

*Hilfestellung:* Die Formel ist irgendwas mit  $\log n$  und dann muss man noch auf- oder abrunden und vielleicht etwas dazu zählen oder abziehen. Machen Sie sich erst mal an Beispielen klar, was genau die Formel ist. Für den Beweis gibt es zahlreiche Möglichkeiten, insbesondere mit und ohne Induktion.

### Aufgabe 3 (10 Punkte)

Sagen Sie in einem Satz, was das folgende Programm berechnet (spielen Sie es dazu für ein paar konkrete natürliche Zahlen  $n \in \mathbb{N}$  durch). Beweisen Sie, dass die Laufzeit höchstens  $C_1 \cdot \log_2 n + C_2$  ist, für Konstanten  $C_1 > 0$ ,  $C_2 > 0$ , die weder von  $n$  noch von  $x$  abhängen.

[logarithmisch umblättern]

In dem folgenden Programm berechnet `% 2` den Rest bei der Division durch 2 (*modulo 2*) und `// 2` ist ganzzahlige Division durch 2 (wie Division durch 2 und dann abgerundet).

```
1. def power(x, n):
2.     result = 1
3.     while n > 0:
4.         if n % 2 == 1:
5.             result = result * x
6.             x = x * x
7.             n = n // 2
8.     return result
```

#### **Aufgabe 4** (Zusatzaufgabe, keine Punkte macht aber Spaß)

Schreiben Sie eine Funktion `majority(array: list[int])`, die für ein gegebenes Feld `array` feststellt, ob es eine Zahl gibt, die echt mehr als  $n/2$  mal vorkommt, wobei  $n$  die Länge des Feldes ist. Falls ja, soll diese (dann eindeutige) Zahl zurückgegeben werden, sonst soll `None` zurückgegeben werden.

Zum Beispiel sollte die Funktion für die Eingabe `[5, 5, 3, 2, 5]` die Zahl 5 zurückgeben und für die Eingabe `[5, 5, 3, 2, 1, 5]` den Wert `None`. Die Funktion soll dabei in Zeit  $\leq C \cdot n$  laufen für eine Konstante  $C > 0$ , die nicht von  $n$  abhängt. Insbesondere haben Sie also nicht genug Zeit, um das Feld zu sortieren.

*Hinweis:* Es reicht, zweimal von links nach rechts über das Feld zu gehen und dabei jeweils eine konstante (von der Eingabegröße  $n$  unabhängige) Anzahl von Variablen auf dem neuesten Stand zu halten. Es ist eine Knobelaufgabe, aber mit diesem Hinweis kann man gut drauf kommen.

Committen Sie Ihre Abgabe als *ein PDF* (bitte Regel 2 vom Wiki beachten) in das SVN, in einem Unterordner `blatt-02`. Auf unserem Wiki finden Sie eine L<sup>A</sup>T<sub>E</sub>X-Vorlage für die Abgabe.

Committen Sie in diesem Unterordner außerdem wie gehabt eine Textdatei `erfahrungen.txt`, in der Sie kurz Ihre Erfahrung mit diesem Übungsblatt und der Vorlesung dazu beschreiben, insbesondere wie lange Sie gebraucht haben und ob es wesentliche Probleme gab und wenn ja welche.

Zu welcher der fünf Gruppen von Folie 5 der zweiten Vorlesung gehören Sie und warum?