

Übungsblatt 4

Abgabe bis Dienstag, den 20. Mai um 12:00 Uhr

Wichtiger Hinweis: sämtliche Aufgaben auf diesem Übungsblatt lassen sich elegant und mit wenig Schreibaufwand lösen. Wenn Sie anfangen, seitenlang zu rechnen, denken Sie zu kompliziert oder übersehen einen einfacheren Weg oder haben etwas noch nicht verstanden. Nutzen Sie die Gelegenheit, solche Beweise zu üben und ein besseres Gefühl dafür zu bekommen, was ein präziser Beweis ist. In der Klausur gibt es auch immer Aufgaben von dieser Art.

Aufgabe 1 (5 Punkte)

Zeigen Sie, dass $\log_a n = \Theta(\log_b n)$ für alle $a, b > 1$. Benutzen Sie dabei für den Beweis direkt die Definitionen aus Vorlesung 4. Sie dürfen für diese Aufgabe *nicht* mit Hilfe von Grenzwerten argumentieren.

Bemerkung: Deswegen schreibt man mit der O-Notation in der Regel einfach nur sowas wie $O(\log n)$ oder $\Theta(\log n)$ oder $\Omega(\log n)$, ohne die Basis des Logarithmus explizit zu nennen.

Aufgabe 2 (5 Punkte)

Sei $f(n) = 2^n$, $g(n) = n!$ und $h(n) = n^n$. Bestimmen Sie, welche der Beziehungen O , Ω , Θ , o , ω jeweils zwischen f , g und h gelten und welche nicht.

Argumentieren Sie dazu über den Grenzwert und seien Sie bei der Bestimmung der Grenzwerte präzise („sieht man doch“ ist kein Beweis).

Aufgabe 3 (5 Punkte)

Seien f_1, f_2, f_3 Funktionen mit $f_1 \in O(f_2)$ und $f_2 \in o(f_3)$. Beweisen Sie, dass $f_1 \in o(f_3)$. Argumentieren Sie über die Definition von O und *nicht* über Grenzwerte.

Bemerkung: Führen Sie die Beweise 100% formal, ohne „hand-waving“.

Aufgabe 4 (5 Punkte)

Bestimmen Sie die Laufzeit der folgenden Funktion als $\Theta(\dots)$ in Abhängigkeit von positiven n (mit Begründung natürlich). Beschreiben Sie außerdem in einem präzisen Satz, was diese Funktion für ein gegebenes n berechnet.

```
1. def bit_counter(n):
2.     bits = [0]
3.     for i in range(0, n):
4.         carry = 1
5.         for j in range(len(bits)):
6.             carry_new = (bits[j] + carry) // 2
7.             bits[j] = (bits[j] + carry) % 2
8.             carry = carry_new
9.         if carry == 1:
10.            bits.append(1)
11.            carry = 0
12.    return bits
```

Committen Sie bitte Ihre Lösungen (als *ein* PDF, siehe Regel 2 auf dem Wiki) sowie Ihre *erfahrungen.txt* (mit der üblichen Rückmeldung zum Übungsblatt und zur Vorlesung dazu) in einen neuen Unterordner *blatt-04*.

Haben Sie Sorge, dass Ihr zukünftiger Job von einer KI übernommen wird?