



Übungsblatt 1

Abgabe via Moodle.
Deadline Fr. 13th Mai

Aufgabe 1 (8 Punkte: 1 + 1 + 1 + 2 + 1 + 2)

Beweisen oder widerlegen Sie:

- a) $n^3 = O(n^3 - 2n^2)$, $n^{\frac{3}{2}} = O(n^2)$, $n^3 = O(10^6 n^2)$.
- b) $2^{n+1} = O(2^n)$, $5 \cos(n) + n = O(1)$, $(n+1)! = O(n!)$.
- c) $\log_2 n = \Theta(\log_{10} n)$, $2^{\log_a(n)} = \Theta(2^{\log_b(n)})$ für $a \neq b$
- d) $f(n)+g(n) = O(\min(f(n), g(n)))$, $f(n)+g(n) = O(\max(f(n), g(n)))$, $f(n)g(n) = O(f(n)g(n))$
- e) $f(n) = O(n) \Rightarrow 2^{f(n)} = O(2^n)$, $f(n) = O(n) \Rightarrow f(n)^2 = O(n^2)$
- f) Auf der Menge der asymptotisch positiven Funktionen ist

$$f \sim_{\Theta} g \text{ gdw. } f \in \Theta(g)$$

eine Äquivalenzrelation.

Aufgabe 2 (Zeitaufwand von Algorithmen, 2 + 1 + 1 + 2 Punkte)

Gehen Sie davon aus, dass Sie mit einem Rechner arbeiten, der nur die Rechenoperationen $+$ und $-$, sowie die Vergleiche $<$, \leq , $=$, $>$, \geq in Zeit $\Theta(1)$ ausführen kann. Zuweisungen an Variablen oder an Einträge von Arrays sind ebenfalls möglich und benötigen ebenfalls $\Theta(1)$ Zeit; ebenso auch das Auslesen von Variablen und Array-Einträgen. Ansonsten unterstützt der Rechner nur die übliche Ablaufsteuerung (Schleifen, Fallunterscheidung, Unterprogrammaufruf) und *keine* weiteren Operationen. Der *einzig*e verfügbare Basisdatentyp ist zudem die Menge $\mathbb{N}_{\geq 0}$. Im folgenden müssen Sie nun einige Algorithmen für diesen Rechner angeben. Dabei spielt nur die Laufzeit eine Rolle. Ihr Algorithmus könnte z.B. immer den gleichen Wert als Ergebnis zurückgeben. Wichtig ist nur, dass er dabei das gewünschte Laufzeitverhalten aufweist.

1. Geben Sie einen Algorithmus an, der als Eingabe eine Zahl $n \in \mathbb{N}_{\geq 0}$ hat und $\Theta((\log n)^2)$ Zeit benötigt.
2. Geben Sie einen Algorithmus an, der als Eingabe eine Zahl $n \in \mathbb{N}_{\geq 0}$ hat und $\Theta(n^2 \log n)$ Zeit benötigt.
3. Geben Sie einen Algorithmus an, der als Eingabe eine Zahl $n \in \mathbb{N}_{\geq 0}$ hat und $\Theta(2^n)$ Zeit benötigt.
4. Geben Sie einen Algorithmus an, der als Eingabe eine Zahl $n \in \mathbb{N}_{\geq 0}$ hat und $\Theta(n^n)$ Zeit benötigt.

Zeigen Sie jeweils, dass Ihr Algorithmus tatsächlich die geforderte Laufzeit hat.

Hinweis: Soweit sinnvoll können Sie dabei die Erkenntnisse aus Aufgabe 1 verwenden.

Aufgabe 3 (Finden einer fehlenden ganzen Zahl, 4 + 1 + 1 Punkte)

Ein Feld $A[0..n-2]$ enthält alle ganzen Zahlen von 0 bis $n-1$ außer einer. In der vorliegenden Problemstellung können wir auf die vollständige ganze Zahl nicht mit einer einzigen Operation zugreifen. Die Elemente von A sind binär dargestellt und die einzige Operation, die wir benutzen können, um auf sie zuzugreifen, ist "rufe das j -te Bit von $A[i]$ auf". Dies benötigt konstante Zeit.

Gehen Sie dabei davon aus, dass $n = 2^k$ für ein $k \in \mathbb{N}$ und die Codierung pro Zahl k Bits verwendet, also führende Nullen vorhanden sind. Weiter können Sie davon vereinfachend davon ausgehen, dass die Zahlen aufsteigend sortiert sind.

1. Geben Sie einen rekursiven Algorithmus an, der die fehlende ganze Zahl immer noch in Zeit $O(n)$ findet.
2. Geben Sie für den Algorithmus eine Rekursion für die Laufzeitabschätzung an.
3. Lösen Sie die Rekursion und beweisen Sie damit, dass ihr Algorithmus in Zeit $O(n)$ läuft.

Aufgabe 4 (Karatsuba-Ofman Multiplikation, 4 Punkte)

Multiplizieren Sie 1231 mit 1622 (das sind Dezimalzahlen!) nach dem Algorithmus von Karatsuba und Ofman.

Hinweis: Führen Sie genau die einstelligen Operationen direkt aus.

Aufgabe P1 (Praktische Aufgabe, optional)

Für die praktischen Übungen verwenden wir die Plattform www.hackerrank.com. Hier müssen Sie sich registrieren um an den Übungen teilzunehmen. Unter dem Link

<https://www.hackerrank.com/ads-i-praktische-uebung>

finden die praktischen Übungen in der Form eines Programmierwettbewerbs statt.

Die erste Challenge dient vor allem dazu, die Plattform HackerRank kennen zu lernen. Die Aufgabe besteht darin ein Programm zu schreiben, das als Eingabe einen Namen bekommt. Die Ausgabe des Programms soll sein: **Hello out there, 'Name'!**. Wobei die Person begrüßt wird, dessen Name das Programm als Eingabe erhält.

Der fertige Code kann direkt über die Plattform eingereicht werden.