Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Dennis Hofheinz
Lukas Barth, Lisa Kohl

1. Übungsblatt zu Algorithmen I im SoSe 2016

https://crypto.iti.kit.edu/index.php?id=algo-sose16
{lisa.kohl,lukas.barth}@kit.edu

Aufgabe 1 ($\mathcal{O}$*-Kalkül,* 8 *Punkte*)

Tragen Sie jeweils zutreffend $\mathcal{O}$, Ω oder Θ in das Kästchen ein und beweisen Sie die entstandene Aussage. Verwenden Sie wann immer möglich Θ.

a) $3n^2 + 14n + 159 \in \boxed{}\,(n^2)$, $\quad 2^n \in \boxed{}\,(n^3)$, $\quad n^2 \log n \in \boxed{}\,(n^3)$

b) $\log n^2 \in \boxed{}\,(\log n^3)$, $\quad \log^2 n \in \boxed{}\left(\log^3 n\right)$, $\quad (n+1)! \in \boxed{}\,(n! + 2^n)$

Beweisen oder widerlegen Sie die folgenden Implikationen.

c) $\forall f, g\colon \mathbb{N} \to \mathbb{N}\colon f(n) \in \mathcal{O}(g(n)) \Rightarrow 2^{f(n)} \in \mathcal{O}\left(2^{g(n)}\right)$

d) $\forall k \in \mathbb{N},\ \forall f, g\colon \mathbb{N} \to \mathbb{N}\colon f(n) \in \mathcal{O}(g(n)) \Rightarrow f(n)^k \in \mathcal{O}\left(g(n)^k\right)$

Aufgabe 2 (*Schleifeninvariante,* 4 *Punkte*)

Gegeben sei folgender Algorithmus in Pseudocode. Gehen Sie dabei vereinfachend davon aus, dass alle auftretenden Zahlen jeweils in eine Speicherzelle passen.

```
Function f(n : ℕ) : ℕ
    a=0 : ℕ
    i=0 : ℕ
    while i < n do
        b=1 : ℕ
        j=0 : ℕ
        while j < i do
            b:= b + b
            j:= j + 1
        a:= a + b
        i:= i + 1
    return a
```

a) Berechnen Sie die Laufzeit des Algorithmus im $\mathcal{O}$-Kalkül.

b) Nennen Sie eine Schleifeninvariante für die innere while-Schleife, d.h. eine Invariante, die vor, nach und am Anfang jedes Schleifendurchlaufes gilt. Diese soll b in Abhängigkeit von j beschreiben.

c) Nennen und beweisen Sie eine Schleifeninvariante für die äußere while-Schleife. Diese soll a in Abhängigkeit von i beschreiben. Was gibt der Algorithmus zurück?

d) Geben Sie einen optimierten Algorithmus an, der auch ausschließlich mit Additionen und Subtraktionen auskommt und die gleiche Funktion berechnet. Geben Sie die verbesserte Laufzeit im $\mathcal{O}$-Kalkül an.

Aufgabe 3 (*Karatsuba-Ofman-Multiplikation,* 4 *Punkte*)

Multiplizieren Sie 1701 mit 8472 (das sind Dezimalzahlen!) nach dem Algorithmus von Karatsuba und Ofman.
Hinweis: Führen Sie nur Additionen sowie einstellige Multiplikationen direkt aus. Mehrstellige Multiplikationen müssen rekursiv mit Karatsuba-Ofman berechnet werden.

Aufgabe 4 (*Algorithmenentwurf,* 3 *Punkte*)

Entwerfen Sie einen Algorithmus für das folgende Problem: Gegeben sei eine Menge M von reellen Zahlen. Eine reelle Zahl $r \in M$ heißt *zusammengesetzt*, wenn sich r als Summe $r = s + t$ zweier Elemente $s, t \in M$ schreiben lässt.

Entwerfen Sie einen Algorithmus in Pseudocode, der für alle $r \in M$ entscheidet, ob diese zusammengesetzt sind. Sie dürfen annehmen, dass M als aufsteigend sortiertes Array vorliegt. Ihr Algorithmus muss asymptotisch Laufzeit $\mathcal{O}(n^2)$ haben. Begründen Sie, warum der Algorithmus diese Laufzeit erreicht.

Ausgabe: Mittwoch, 20.04.2016
Abgabe: Freitag, 29.04.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 1
Algorithmen I

Tutoriumsnummer:

Name	**Matrikelnummer**	**Unterschrift**

Mit unseren Unterschriften bestätigen wir, dass die Aufgaben von den Unterzeichnern eigenständig gelöst worden sind.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie in das obere Quadrat *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.