Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Dennis Hofheinz
Lukas Barth, Lisa Kohl

2. 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 (*Rekurrenzen, 6 Punkte*)

a) Gegeben sei folgende Rekurrenz:

$$T(n) = \begin{cases} 1 & \text{für } n < 3, \\ 9 \cdot T\left(\frac{n}{3}\right) + 7n & \text{für } n \geq 3 \end{cases}$$

Zeigen Sie durch vollständige Induktion, dass $T(n) \leq 47n^2 - 4n$ gilt, falls sich n als $n = 3^k$ oder $n = 2 \cdot 3^k$ (mit $k \in N_0$) schreiben lässt.

b) Gegeben sei folgende Rekurrenz:

$$T(n) = \begin{cases} c_0 n & \text{falls } n \leq 20, \\ T(\lceil \frac{n}{4} \rceil) + T(\left\lceil \frac{5}{12}n + \frac{3}{2} \right\rceil) + c_1 n & \text{falls } n > 20. \end{cases}$$

Finden Sie eine Funktion f, so dass $T(n) \in \Theta(f(n))$ gilt und beweisen Sie Ihre Behauptung.

Aufgabe 2 (*Master-Theorem, 4 Punkte*)

Zeigen Sie mit Hilfe der gerundeten Version des Master-Theorems scharfe asymptotische Schranken für folgende Rekurrenzen:

a) $A(1) = 1$ und für $n \in \mathbb{N}$: $A(n) = 5A(\lceil n/5 \rceil) + 12n$

b) $B(1) = 6$ und für $n \in \mathbb{N}$: $B(n) = 8B(\lceil n/2 \rceil) + 4n + 3\sqrt{n} + 17$

c) $C(1) = 12$ und für $n \in \mathbb{N}$: $C(n) = C(\lceil n/7 \rceil) + n$

d) $D(1) = 5$ und für $n \in \mathbb{N}$: $D(n) = 7D(\lceil n/3 \rceil) + C(n) + 7n$

Aufgabe 3 (*$\mathcal{O}$-Kalkül, 3 Punkte*)

Sortieren Sie die folgenden Funktionen von der asymptotisch kleinsten zur asymptotisch größten. Schreiben Sie $f(n) \ll g(n)$, falls $f(n) \in o(g(n))$ und $f(n) \equiv g(n)$ falls $f(n) \in \Theta(g(n))$. Es werden keine Beweise benötigt.

$n!$	$\log_n n$	$\log n$	n^n
$n^{1,001}$	π	$\log^{\sqrt{n}} n$	$n/\log n$
$\log_{1000} n$	10^n	n	$n \log n$
$n^{\log n}$			

Aufgabe 4 (*Schleifeninvarianten,* 6 *Punkte*)

Für zwei gegebene ganze Zahlen $a, b \in \mathbb{N}_0$ heißt eine Zahl $m \in \mathbb{N}_0$ der *größte gemeinsame Teiler* (greatest common divisor, GCD) von a und b, wenn

(i) m gemeinsamer Teiler von a und b ist, in Zeichen $m \mid a$ und $m \mid b$, und

(ii) für *jeden* gemeinsamen Teiler $m' \in \mathbb{N}_0$ aus $m' \mid a$ und $m' \mid b$ unmittelbar $m' \mid m$ folgt.

Der *Euklidische Algorithmus* berechnet zu zwei gegebenen Zahlen $a, b \in \mathbb{N}_0$ den größten gemeinsamen Teiler. Wir betrachten im Folgenden drei Varianten des Grundalgorithmus.

Zeigen Sie für jede der drei Varianten die Korrektheit, indem sie an den benötigten Stellen Assertions und Invarianten einfügen und diese beweisen. Falls notwendig, fügen Sie Hilfsvariablen ein oder annotieren Sie Variablen mit Indizes, um den entsprechenden Schleifendurchlauf zu bezeichnen.

a) Übersetzt man die alten Schriften von Euklid in heutigen Pseudocode, dann sieht dieser Algorithmus wie folgt aus:

Function euklid($a : \mathbb{N}_+$; $b : \mathbb{N}_+$) : $\mathbb{N}_+$
    while $a \neq b$ **do**
        if $a > b$ **then** swap(a, b)
        $b := b - a$
    return a

b) Die wiederholten Subtraktionen kann man durch eine einzige Division mit Rest ersetzen:

Function euklidWithModulo($a : \mathbb{N}$; $b : \mathbb{N}$) : $\mathbb{N}$
    if $a < b$ **then** swap(a, b)
    while $b \neq 0$ **do**
        $(a, b) := (b, a \textbf{ mod } b)$
    return a

c) In der Kryptographie ist der sogenannte *Erweiterte* Euklidische Algorithmus von besonderer Bedeutung. Er berechnet neben dem GCD noch zwei Zahlen $s, u \in \mathbb{Z}$, sodass sich $m = \gcd(a, b)$ schreiben lässt als

$$m = a \cdot s + b \cdot u\,.$$

Die Idee hinter der Erweiterung besteht darin bei der Division mit Rest, die ganzzahligen Vielfache, a **div** b, nicht zu verwerfen, sondern in s und u zu sammeln. Die Zahlen s und u kann man *nach* Ausführen von euklidWithModulo() durch Rückeinsetzen der Divisionen errechnen, oder, noch geschickter, *währenddessen*:

Function extendedEuklid($a : \mathbb{N}$; $b : \mathbb{N}$) : $(\mathbb{N}, \mathbb{Z}, \mathbb{Z})$
    $(s, t) := (1, 0)$
    $(u, v) := (0, 1)$
    while $b > 0$ **do**
        $q := a$ **div** b
        $(a, b) := (b, a - qb)$
        $(s, t) := (t, s - qt)$
        $(u, v) := (v, u - qv)$
    return (a, s, u)

Hinweis: Es gilt $\gcd(a, 0) = \gcd(0, a) = a$.

Aufgabe 5 (*Graphen,* 3 *Punkte*)

Der ebenso brilliante wie einflussreiche Wissenschaftler und Superbösewicht Doktor Meta hat sich ein internationales Netzwerk von Helfern und hochrangigen Kontakten aufgebaut. In diesem Netzwerk ist er nicht mit allen persönlich bekannt, baut aber darauf, seinen Einfluss auch über Mittelspersonen ausüben zu können. Da er von Grund auf höchst misstrauisch ist und mögliche Kollaborationen gegen seine Person verhindern möchte, hat er peinlich genau darauf geachtet, dass es im Netzwerk keine drei Personen gibt, die sich alle direkt kennen. Gleichzeitig ist er für ein stabiles Netzwerk an einer möglichst hohe Dichte der Bekanntschaften interessiert.
Zeigen Sie, dass in diesem Szenario bei $2n$ Personen im Netzwerk (inklusive Doktor Meta) höchstens n^2 Bekanntschaften möglich sind. Zeigen sie weiterhin, dass es für jedes $n \in \mathbb{N}$ auch ein Netzwerk mit n^2 Bekanntschaften gibt, das diese Bedingung erfüllt.

Ausgabe: Mittwoch, 27.04.2016
Abgabe: Freitag, 06.05.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 2
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.