Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Komplexität

186.813 Algorithmen und Datenstrukturen 1 VU 6.0

Übungsblatt 1

für die Übung am Montag den 20. bzw. Dienstag den 21. März 2017.

Geben Sie bis **spätestens Sonntag, 19.03.2017, 12:00 Uhr** über TUWEL an, welche Beispiele Sie bearbeitet und gelöst haben. Gehen Sie dabei folgendermaßen vor:

- TUWEL (`https://tuwel.tuwien.ac.at`)
  Kurs *186.813 Algorithmen und Datenstrukturen 1 (VU 6.0)*
- Übungsblätter
- Bearbeitete Beispiele ankreuzen **und** abgeben
  - Link *Übungsblatt 1 - Details & Bewertung*
    Button *Abgabe bearbeiten*
    Bearbeitete Beispiele anhaken und *Änderungen speichern*.
  - Link *Hochladen Lösungen Übungsblatt 1*
    Button *Abgabe hinzufügen*
    PDF-Datei mit Lösungen hochladen und *Änderungen sichern*.

Bitte beachten Sie:

- Sie können **vor** der Deadline beliebig oft ihre Auswahl an Beispielen und das zugehörige Lösungs-PDF verändern, aber **nach** der Deadline gibt es **keine** Veränderung ihrer angekreuzten Beispiele **und** der PDF-Datei!
- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen einreichen.
- Bitte geben Sie Ihren Namen, Matrikelnummer und E-Mail-Adresse in den Ausarbeitungen an.
- Wenn Sie zur Präsentation Ihrer Lösung eines von Ihnen angekreuzten Beispiels ausgewählt werden und dieses aber nicht bearbeitet haben oder dieses Beispiel nicht in der PDF-Datei vorhanden ist, verlieren Sie **alle** Punkte dieser Übungseinheit!

  (Zusätzlich werden stichprobenartig die abgegebenen PDF-Dateien auf Übereinstimmung mit der entsprechenden Kreuzerlliste überprüft.)

Aufgabe 1 Betrachten Sie folgende Problemstellung für das Stable-Matching-Problem mit, z. B., vier Frauen (A,B,C,D) und vier Männern (W,X,Y,Z). Berechnen Sie ein Stable Matching mithilfe des Gale-Shapley-Algorithmus aus den Vorlesungsfolien. Lässt der Algorithmus offen, welche/r Frau/Mann als nächstes betrachtet werden soll, dann gehen Sie in alphabetischer Reihenfolge vor. Stellen Sie die einzelnen Berechnungsschritte in Form der aktuell fixierten Paare dar (alle Zwischenschritte sollen angegeben werden).

	1.	2.	3.	4.
W	B	D	A	C
X	C	A	B	D
Y	B	A	D	C
Z	A	B	D	C

	1.	2.	3.	4.
A	W	Z	X	Y
B	Z	W	X	Y
C	Z	X	Y	W
D	X	Z	Y	W

Aufgabe 2 Ordnen Sie folgende Funktionen nach Dominanz, beginnend mit der asymptotisch am schwächsten wachsenden. Kennzeichnen sie gesondert Funktionen, die gleich schnell wachsen.

$$\log\left(n^{12}\right), \quad \left(\frac{9}{8}\right)^n, \quad 2n^{10} - 8n^4, \quad \left(\frac{5}{6}\right)^n, \quad \sqrt{n^{16}}, \quad n^2 \left(\log n\right)^2, \quad n^5 \cdot 5 \cdot n^5$$

(Es genügt die korrekte Reihung anzugeben, ein formeller Beweis ist nicht erforderlich. Sie müssen aber in der Lage sein, Ihre Lösung zu erklären.)

Aufgabe 3 Gegeben ist die folgende Funktion:

$$f(n) = \begin{cases} 2^n \cdot \left(\frac{6}{4}\right)^n + 3^n \cdot 3 \log_3 3^n, & \text{falls } (n < 10^3) \text{ oder } (n \geq 10^5) \\ n \log_6 n + 6n, & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$\Theta(.)$	$O(.)$	$\Omega(.)$	keines
$3^n \log_2 n$				
$n!$				
$n \log n$				
$n3^n$				

Aufgabe 4 Beantworten Sie folgende Fragen und begründen Sie Ihre Antwort:

- Wenn die Worst-Case-Laufzeit eines Algorithmus in $\Theta(n^2)$ liegt, ist es dann möglich, dass seine Laufzeit für **alle** Instanzen in $O(n)$ liegt?
- Wenn die Worst-Case-Laufzeit eines Algorithmus in $\Theta(n^2)$ liegt, ist es dann möglich, dass seine Laufzeit für **manche** Instanzen in $O(n)$ liegt?

Aufgabe 5 Im Folgenden seien zwei Arrays A, B mit jeweils n Elementen gegeben:

- A: Jedes Element an einer ungeraden Position ist größer als jedes Element an einer geraden Position (z.B. $4, 6, 2, 8, 1, 7, 3, 5$)
- B: Ausgehend von einem aufsteigend sortierten Array wird jedes Element an einer Position, die durch 3 teilbar ist, mit dem direkt darauffolgenden Element vertauscht (z.B. $2, 1, 3, 5, 4, 6, 8, 7$)

Geben Sie für jedes Array die Laufzeit von Insertion-Sort (gemäß dem Pseudocode aus den Vorlesungsfolien) in Θ-Notation an:

Array	$\Theta(\cdot)$
A	
B	

Nehmen Sie an, dass Arrays immer mit 0 beginnend indiziert werden. 0 gilt als gerade und durch 3 teilbar.

Aufgabe 6 Bestimmen Sie für den gegebenen Pseudocode

- die **Laufzeit** in Abhängkeit von n in Θ-Notation, und
- die **Werte der Variablen** a **und** b nach dem Ausführen des Codes in Abhängkeit von n in Θ-Notation.

$a \leftarrow 1$
$b \leftarrow 1$
for $c \leftarrow 1, \ldots, \lfloor\sqrt{n}\rfloor$
    $a \leftarrow a + a$
    $b \leftarrow b + c$
$i \leftarrow 2^{2b}$
while $i > 1$
    $i \leftarrow \lfloor\frac{i}{2}\rfloor$

Aufgabe 7 Bestimmen Sie die Laufzeit für den folgenden Pseudocode in Abhängigkeit von n in Θ-Notation.

$i \leftarrow 1$
$j \leftarrow n$
$k \leftarrow 1$
while $i < j$
    $k \leftarrow k + n$
    $j \leftarrow \left\lfloor \frac{j}{2} \right\rfloor$
$l \leftarrow 1$
for $i \leftarrow 1, \ldots, k$
    for $j \leftarrow 1, \ldots, i$
        $l \leftarrow l + 1;$
$j \leftarrow n$
for $i \leftarrow 1, \ldots, l$
    $j \leftarrow 2 \cdot j + i$

Aufgabe 8 Beweisen Sie: $f(n) + g(n) = O\left(\max(f(n), g(n))\right)$.
