

186.813 Algorithmen und Datenstrukturen 1 VU 6.0

Übungsblatt 1

für die Übung am Montag den 14. bzw. Dienstag den 15. März 2016.

Geben Sie bis **spätestens Sonntag, 13.03.2016, 23:59 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 anhängen 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 vier Frauen (A-D) und vier Männern (W-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.		1.	2.	3.	4.
W	B	D	A	C	A	W	Z	X	Y
X	C	A	B	D	B	Z	W	X	Y
Y	B	A	D	C	C	Z	X	Y	W
Z	A	B	D	C	D	X	Z	Y	W

Aufgabe 2 Gegeben seien drei Algorithmen mit den unten angeführten Laufzeiten. Nehmen Sie an, dass es sich dabei um die exakte Anzahl an ausgeführten Operationen in Abhängigkeit der Eingabegröße n handelt. Ihnen steht ein Computer zur Verfügung, der 10^{12} Operationen pro Sekunde ausführen kann. Wie groß ist die maximale Eingabegröße n , für die Sie innerhalb einer Stunde auf besagter Hardware ein Ergebnis berechnen können?

- $100 \cdot n^2$
- $10000 \cdot 2^n$
- 2^{2^n}

Aufgabe 3 Gegeben seien zwei Funktionen $f(n)$ und $g(n)$ für die gilt, dass $f(n) = O(g(n))$ und $g(n) = O(f(n))$. Beweisen Sie nachfolgende Aussagen oder widerlegen Sie sie durch Angabe eines Gegenbeispiels:

- $f(n) = \Theta(g(n))$
- $f(n) = g(n)$

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

- Ist es möglich, dass ein Algorithmus auf **bestimmten** Instanzen eine Laufzeit von $O(n)$ aufweist, wenn seine Worst-Case-Laufzeit in $O(n^2)$ liegt?
- Ist es möglich, dass ein Algorithmus auf **allen** Instanzen eine Laufzeit von $O(n)$ aufweist, wenn seine Worst-Case-Laufzeit in $O(n^2)$ liegt?

Aufgabe 5 Füllen Sie folgende Tabelle aus und begründen Sie Ihre Antworten:

	richtig	falsch
$3^n = O(2^n)$		
$\log 3^n = O(\log 2^n)$		
$3^n = \Omega(2^n)$		
$\log 3^n = \Omega(\log 2^n)$		

Aufgabe 6 Finden Sie für nachfolgende Funktionen f_i jeweils möglichst einfache Funktionen g_i , sodass gilt $f_i(n) = \Theta(g_i(n))$ ($i \in \{1, 2, 3, 4\}$).

- $f_1(n) = 1000 \cdot 2^n + 4^n$
 - $f_2(n) = n + n \cdot \log n + \sqrt{n}$
 - $f_3(n) = \log_2(n^{20}) + (\log_2 n)^{10}$
 - $f_4(n) = 0.99^n + n^{100}$
-

Aufgabe 7 Bestimmen Sie die Laufzeit des folgenden Pseudocodes in O -Notation in Abhängigkeit von der Eingabegröße n . Geben Sie des Weiteren den Rückgabewert (r) als Funktion in Abhängigkeit der Eingabegröße n an.

```
mystery():  
   $r \leftarrow 0$   
  for  $i \leftarrow 1$  bis  $n - 1$   
    for  $j \leftarrow i + 1$  bis  $n$   
      for  $k \leftarrow 1$  bis  $j$   
         $r \leftarrow r + 1$   
  return  $r$ 
```

(Hinweis: $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$)

Aufgabe 8 Beweisen Sie folgende Aussage: Falls $f_1(n) = O(g_1(n))$ und $f_2(n) = O(g_2(n))$ dann gilt $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$.
