TU Wien
Institut für Logic and Computation
Forschungsbereich Algorithms and Complexity

186.866 Algorithmen und Datenstrukturen VU

Übungsblatt 1

PDF erstellt am: 15. März 2023

Deadline für dieses Übungsblatt ist **Montag, 27.03.2023, 20:00 Uhr**. Um Aufgaben für diese Übung anerkannt zu bekommen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den TUWEL-Kurs der Lehrveranstaltung *186.866 Algorithmen und Datenstrukturen (VU 5.5)* und navigieren Sie zum Abschnitt *Übungsblätter*.
2. Teilen Sie uns mit, welche Aufgaben Sie gelöst haben **und** welche gelösten Aufgaben Sie gegebenenfalls in der Übungseinheit präsentieren können. Gehen Sie dabei folgendermaßen vor:
   - Laden Sie Ihre Lösungen in einem einzigen PDF-Dokument in TUWEL hoch.
Link *Hochladen Lösungen Übungsblatt 1*
Button *Abgabe hinzufügen*
PDF-Datei mit Lösungen hochladen und *Änderungen sichern.*
   - Kreuzen Sie an, welche Aufgaben Sie gegebenenfalls in der Übung präsentieren können. Die Lösungen der angekreuzten Aufgaben müssen im hochgeladenen PDF enthalten sein.
Link *Ankreuzen Übungsblatt 1*
Button *Abgabe bearbeiten*
Bearbeitete Aufgaben anhaken und *Änderungen speichern.*

Bitte beachten Sie:

- Bis zur Deadline können Sie sowohl Ihr hochgeladenes PDF, als auch Ihre angekreuzten Aufgaben beliebig oft verändern. Nach der Deadline ist keine Veränderung mehr möglich. Es werden ausnahmslos keine Nachabgabeversuche (z.B. per E-Mail) akzeptiert.
- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen hochladen (beachten Sie die maximale Dateigröße).
- Beachten Sie die Richtlinien für das An- und Aberkennen von Aufgaben (Details finden Sie in den Folien der Vorbesprechung).

Aufgabe 1. Ordnen Sie folgende Funktionen nach Dominanz, beginnend mit der asymptotisch am schwächsten wachsenden. Es genügt die Funktionen zu reihen, ein Beweis der Gültigkeit der Relationen ist nicht erforderlich.

$$3\frac{n^{2.5}}{n^2}, \quad 5\log(n)^2, \quad \frac{n^2}{10}, \quad \frac{n!}{2n^2}, \quad 2^{n+5}, \quad \left(\frac{4}{5}\right)^n, \quad 3^{n/2}, \quad \log(n^{5/3})$$

Aufgabe 2. Gegeben ist die folgende Funktionen:

$$f(n) = 2^n(100n^2 + 20n + 7)$$

$$g_1(n) = \begin{cases} 100n^2 + 2^n \cdot n^2 + 7n + (\frac{3}{2})^n & \text{falls } (n < 10^3) \text{ oder } (n \geq 10^5) \\ 5^n + 2^{\frac{3n}{2}} + 100n & \text{sonst} \end{cases}$$

$$g_2(n) = \frac{2^{2^n}}{10000}$$

$$g_3(n) = \begin{cases} 60 & \text{falls } n \text{ eine Primzahl ist} \\ n^{n^n} & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an und begründen Sie Ihre Antworten:

$f(n)$ ist in	$\Theta(.)$	$O(.)$	$\Omega(.)$	keines
$g_1(n)$				
$g_2(n)$				
$g_3(n)$				

Hinweis: Setzten Sie statt dem Punkt die entsprechend Funktion (g_1, g_2 bzw. g_3) ein. Beispielsweise ist die Zelle links oben als „$f(n)$ ist in $\Theta(g_1(n))$" zu lesen.

Aufgabe 3. Beweisen oder widerlegen Sie die folgenden Aussagen:

(a) $\max(f(n), g(n)) = \Theta(f(n) + g(n))$

(b) $(n + a)^b = \Theta(n^b)$ für beliebige ganze Zahlen a und b, wobei $b > 0$.
Hinweis: Verwenden Sie Induktion über b.

(c) Die Best-Case und Worst-Case Laufzeit eines Algorithmus ist $\Theta(g(n))$ genau dann, wenn seine Worst-Case Laufzeit $O(g(n))$ und seine Best-Case Laufzeit $\Omega(g(n))$ ist.

Anmerkung: Wir nehmen an, dass alle Funktionen die nicht-negativen ganzen Zahlen als Definitions- und Wertebereich haben.

Aufgabe 4. Bestimmen Sie die Worst-Case und Best-Case Laufzeiten der unten angegebenen Algorithmen in Abhängigkeit von n in Θ-Notation.

(a)

$s \leftarrow 0$
for $i = 0, \ldots, n-1$
  if $A[i] \bmod 2 = 0$ **then**
    $s \leftarrow s + A[i]$
return s

(b)

$s \leftarrow 0$
for $i = 0, \ldots, n-1$
  if $A[i-1] =$ **True then**
    for $j = 0, \ldots, i$
      $s \leftarrow s + B[j]$
return s

(c)

$a \leftarrow n$
$d \leftarrow 0$
while $a > 0$
  $d \leftarrow d + 1$
  $a \leftarrow \lfloor a/10 \rfloor$
return d

(d)

$s \leftarrow 0$
for $i = 0, \ldots, n-1$
  for $j = 0, \ldots, i$
    for $k = 0, \ldots \lfloor j/2 \rfloor$
      $s \leftarrow s + A[k]$
return s

Aufgabe 5. Bestimmen Sie die Laufzeiten und die Werte der Variablen a und b nach der Ausführung der unten angegebenen Algorithmen in Abhängigkeit von n in Θ-Notation. Verwenden Sie hierfür möglichst einfache Terme.

(a)

$a \leftarrow 1$
$b \leftarrow 1$
for $c = 1, \ldots, \lfloor \frac{1}{2} \log_2 n \rfloor$
  $a \leftarrow 4a$
  $b \leftarrow c$
  $a \leftarrow \lfloor \frac{a}{2} \rfloor$
$i \leftarrow 2^{2b}$
while $i > 1$
  $i \leftarrow \lfloor \frac{i}{2} \rfloor$

(b)

$a \leftarrow n^3;$
$b \leftarrow 1;$
while $a > 1$
  $b \leftarrow b + 1;$
  $c \leftarrow n;$
  do
    $a \leftarrow \lfloor \frac{b}{2} \rfloor;$
    $c \leftarrow \frac{2c}{n};$
    $b \leftarrow b \cdot n;$
  while $c \geq n$

Aufgabe 6. Bestimmen Sie die Laufzeit der unten angegebenen Funktion $f(A,n)$ in Abhängigkeit von n in Θ-Notation (A ist ein Array der Länge n).

```
Function f(A, n):
    if n = 0 then
        return 0
    s ← 0
    for i = 0, ..., n − 1
        s ← s + A[i]
    s ← s + f(A, ⌊n/2⌋)
    return s
```
