



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2012

26. April 2012

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:	<input type="text"/>	Vorname:	<input type="text"/>
Matrikelnummer:	<input type="text"/>	Studienkennzahl:	<input type="text"/>
Anzahl abgegebener Zusatzblätter:	<input type="text"/>	Unterschrift:	<input type="text"/>

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.
Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechner, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	16	16	18	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Erfolg!

Aufgabe 1.A: $\Omega/O/\Theta$ -Notation**(16 Punkte)**

a) (10 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} \frac{n}{3^n} + \frac{1}{\sqrt{n}} + \log_3 n, & \text{wenn } n > 1000 \text{ und gerade} \\ \frac{n}{\sqrt{5^n}} + \log_2(n^3), & \text{wenn } n > 1000 \text{ und prim} \\ 4n \cdot \log_2(4^n), & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$O(\cdot)$	$\Omega(\cdot)$	$\Theta(\cdot)$	keines
$n^2 \log n$				
n				
$\log n$				
n^2				
\sqrt{n}				

Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

b) (6 Punkte)

Beweisen oder widerlegen Sie, dass für beliebige positive Funktionen $f(n)$ und $g(n)$ die folgende Beziehung gilt:

$$g(n) = \Omega(f(n)) \quad \Rightarrow \quad g(n) = O(n^2 \cdot f(n))$$

Aufgabe 3.A: Datenstrukturen

(18 Punkte)

Gegeben sind zwei doppelt verkettete, azyklische Listen A und B .

a) (16 Punkte)

Schreiben Sie in detailliertem Pseudocode eine Funktion `split_and_concat(A, x, B)`, welche die ersten x (mit $x \geq 1$) Elemente aus der Liste A entfernt und **vorne** an die Liste B hängt.

b) (2 Punkte)

Geben Sie die Laufzeit Ihres Algorithmus in Θ -Notation an.

Beachten Sie folgende Punkte:

- Sie können davon ausgehen, dass die Listen nicht leer sind.
- Die Zeiger `A.begin` bzw. `B.begin` verweisen auf das erste Element der Liste A bzw. B .
- Ein Listenelement a speichert neben verschiedenen anderen Daten jeweils einen Zeiger auf seinen Vorgänger (`a.pred`) und Nachfolger (`a.next`). Für das erste bzw. letzte Element gilt `a.pred = NULL` bzw. `a.next = NULL`.
- Wenn die Liste A aus weniger als $x + 1$ Elementen besteht, soll der Algorithmus eine Fehlermeldung ausgeben und abbrechen.
- Die Länge einer Liste kann nur durch Abzählen der Elemente bestimmt werden.



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2012

26. April 2012

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:	<input type="text"/>	Vorname:	<input type="text"/>
Matrikelnummer:	<input type="text"/>	Studienkennzahl:	<input type="text"/>
Anzahl abgegebener Zusatzblätter:	<input type="text"/>	Unterschrift:	<input type="text"/>

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.
Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechner, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

	B1:	B2:	B3:	Summe:
Erreichbare Punkte:	16	16	18	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Glück!

Aufgabe 1.B: Sortierverfahren und Datenstrukturen (16 Punkte)

In der Vorlesung haben Sie das Sortierverfahren Heapsort kennengelernt, mit dem unter Zuhilfenahme eines Maximum-Heaps eine beliebige Folge aufsteigend sortiert werden kann. Analog dazu kann ein Minimum-Heap für absteigendes Sortieren verwendet werden. In diesem Fall wird das jeweilige Minimum mit der letzten Stelle der betrachteten Folge vertauscht.

Definition: Wir nennen eine Folge $F = \langle k_1, k_2, \dots, k_n \rangle$ von Schlüsseln einen *Minimum-Heap*, wenn für alle $i \in \{2, 3, \dots, n\}$ gilt $k_i \geq k_{\lfloor \frac{i}{2} \rfloor}$. Anders ausgedrückt: Falls $2i \leq n$ bzw. $2i + 1 \leq n$, dann muss gelten: $k_i \leq k_{2i}$ bzw. $k_i \leq k_{2i+1}$.

- (6 Punkte) Erfüllen die folgenden Zahlenfolgen die Eigenschaften eines Maximum- bzw. Minimum-Heaps? Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an. Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

	Maximum-Heap	Minimum-Heap	keines
1, 2, 7, 5, 3, 8, 9, 10, 6, 4			
10, 6, 9, 4, 5, 8, 3, 7, 2, 1			
10, 9, 8, 7, 4, 1, 3, 6, 2, 5			
1, 2, 5, 7, 3, 8, 9, 10, 6, 4			
10, 9, 4, 7, 8, 1, 3, 6, 2, 5			
1, 3, 2, 5, 8, 10, 4, 7, 6, 9			

- (10 Punkte) Führen Sie auf die Zahlenfolge $\langle 7, 5, 4, 10, 3, 8, 2, 1, 9, 6 \rangle$ den Algorithmus Heapsort aus, um die Zahlenfolge unter Verwendung eines Minimum-Heaps absteigend zu sortieren. Geben sie das Feld nach dem Aufruf von `ErstelleHeap()` und dann nach jedem Versickerungsschritt an (die graphische Darstellung des Heaps reicht *nicht*).

	7, 5, 4, 10, 3, 8, 2, 1, 9, 6
nach <code>ErstelleHeap()</code>	

Aufgabe 2.B: $\Omega/O/\Theta$ -Notation**(16 Punkte)**

a) (10 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} \frac{n}{\sqrt{5^n}} + \log_2(n^3), & \text{wenn } n > 1000 \text{ und prim} \\ \frac{1}{\sqrt{n}} + \frac{n}{3^n} + \log_3 n, & \text{wenn } n > 1000 \text{ und gerade} \\ 4n \cdot \log_2(4^n), & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$O(\cdot)$	$\Omega(\cdot)$	$\Theta(\cdot)$	keines
n				
$\log n$				
\sqrt{n}				
$n^2 \log n$				
n^2				

Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

b) (6 Punkte)

Beweisen oder widerlegen Sie, dass für beliebige positive Funktionen $f(n)$ und $g(n)$ die folgende Beziehung gilt:

$$g(n) = O(f(n)) \quad \Rightarrow \quad n^2 \cdot g(n) = \Omega(f(n))$$

Aufgabe 3.B: Datenstrukturen

(18 Punkte)

Gegeben sind zwei doppelt verkettete, azyklische Listen A und B .

a) (16 Punkte)

Schreiben Sie in detailliertem Pseudocode eine Funktion `split_and_concat(A, x, B)`, welche die ersten x (mit $x \geq 1$) Elemente aus der Liste A entfernt und **vorne** an die Liste B hängt.

b) (2 Punkte)

Geben Sie die Laufzeit Ihres Algorithmus in Θ -Notation an.

Beachten Sie folgende Punkte:

- Sie können davon ausgehen, dass die Listen nicht leer sind.
- Die Zeiger `A.begin` bzw. `B.begin` verweisen auf das erste Element der Liste A bzw. B .
- Ein Listenelement a speichert neben verschiedenen anderen Daten jeweils einen Zeiger auf seinen Vorgänger (`a.pred`) und Nachfolger (`a.next`). Für das erste bzw. letzte Element gilt `a.pred = NULL` bzw. `a.next = NULL`.
- Wenn die Liste A aus weniger als $x + 1$ Elementen besteht, soll der Algorithmus eine Fehlermeldung ausgeben und abbrechen.
- Die Länge einer Liste kann nur durch Abzählen der Elemente bestimmt werden.