



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2014

30. April 2014

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:

Vorname:

Matrikelnummer:

Unterschrift:

Anzahl abgegebener Zusatzblätter:

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. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte!).

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

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	16	18	16	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) (8 Punkte)

Gegeben sei die folgende Funktion $f : \mathbb{N}^+ \rightarrow \mathbb{R}$

$$f(n) = \begin{cases} e^n \cdot \sin^2(n) + \frac{1}{2 \ln(n)} + n & \text{wenn } n > 42 \text{ und gerade} \\ e \cdot \sin^2(n) + \frac{3}{4}n^2 - \ln(n) & \text{wenn } n \leq 42 \\ n^{-3} + e^{n+1} \cdot \sin^2(n) + \ln(n) & \text{sonst} \end{cases}$$

Gehen Sie davon aus, dass das Argument von $\sin()$ in Grad angegeben wird. Kreuzen Sie **in der folgenden Tabelle** die zutreffenden Felder an:

$f(n)$ ist	$O(\cdot)$	$\Omega(\cdot)$	$\Theta(\cdot)$	keines
e^n				
2^n				
$\sin^2(n)$				
n				

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

b) (8 Punkte)

Tragen Sie für die Codestücke A und B jeweils die **Laufzeit** in Abhängigkeit von n und die **Werte der Variablen a und b** nach dem Ausführen des Codes in Θ -Notation in die unten stehenden Tabellen ein.

A $a = 1;$
 $b = 1;$
für $c = 1, \dots, \frac{1}{2} \log_2 n$ {
 $a = 4a;$
 $b = c;$
 $a = \lfloor \frac{a}{2} \rfloor;$
}
 $i = 2^{2b};$
solange $i > 1$ {
 $i = \lfloor \frac{i}{2} \rfloor;$
}

B $a = n^3;$
 $b = 1;$
solange $a > 1$ {
 $b = b + 1;$
 $c = n;$
wiederhole
 $a = \lfloor \frac{b}{2} \rfloor;$
 $c = \frac{2c}{n};$
 $b = b \cdot n;$
bis $c < n$
}

Laufzeit	a	b
$\Theta(\quad)$	$\Theta(\quad)$	$\Theta(\quad)$

Laufzeit	a	b
$\Theta(\quad)$	$\Theta(\quad)$	$\Theta(\quad)$

Aufgabe 2.A: Sortierverfahren

(18 Punkte)

Gegeben sei der folgende Algorithmus **Sortiere**(A, n) in Pseudocode, der für ein Feld $A = (A[1], \dots, A[n])$ aufgerufen wird:

Sortiere(A, n)

wiederhole

vertauscht = *falsch*;

für $i = 1, 2, \dots, n - 1$ {

falls $A[i + 1] \geq A[i]$ **dann** {

vertausche $A[i + 1]$ und $A[i]$;

vertauscht = *wahr*;

}

}

bis *vertauscht* == *falsch*;

- (7 Punkte)

Überlegen Sie sich die Funktionsweise anhand der Eingabefolge

$$A = \langle 3, 2, 8, 5, 1, 6 \rangle$$

Wenden Sie **Sortiere**(A, n) auf diese Folge an und schreiben Sie dabei den Zustand der Eingabefolge zu Beginn jedes Durchlaufs der **wiederhole**-Schleife auf.

Beschreiben Sie anschließend mit wenigen Worten die **Funktionsweise**.

- (4 Punkte)

Algorithmus **Sortiere**(A, n) arbeitet nicht für jede beliebige Zahlenfolge korrekt. Beschreiben Sie in wenigen Worten, was genau das Problem ist und wie es behoben werden kann.

- (5 Punkte)

Bestimmen Sie die Laufzeit im Worst-Case der korrigierten Version von **Sortiere**(A, n) in Θ -Notation in Abhängigkeit der Eingabegröße n .

Geben Sie anschließend eine Folge mit vier Zahlen an, bei der die Worst-Case Laufzeit zustande kommt.

- (2 Punkte)

Ist der korrigierte Algorithmus stabil? Begründen Sie Ihre Antwort!

Aufgabe 3.A: Binäre Suchbäume

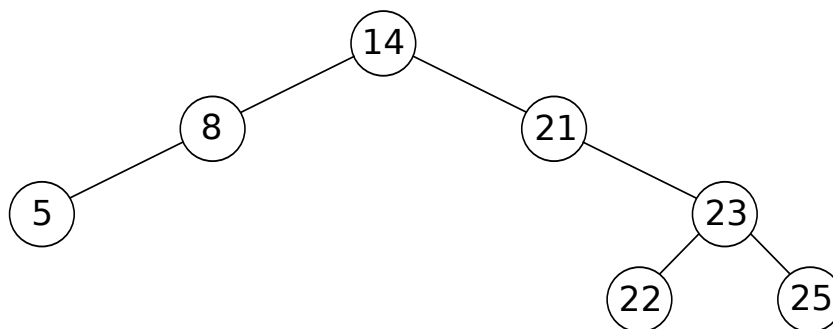
(16 Punkte)

a) (7 Punkte)

Gegeben ist folgender binärer Suchbaum.

- Fügen Sie die Elemente 17, 19 und 10 in den Baum ein.
- Löschen Sie anschließend die Elemente 17 und 14.
- Fügen Sie danach die Elemente 20 und 17 ein.

Falls Sie einen Ersatzknoten benötigen, verwenden Sie den Successor. Führen Sie alle Operationen in der angegebenen Reihenfolge aus und zeichnen Sie den Baum nach jeder einzelnen Einfüge- bzw. Löschoperation.



b) (9 Punkte)

Bei den folgenden *Post-Order* Traversierungen binärer Suchbäume ist jeweils ein Fehler unterlaufen. Finden Sie für jeden Baum den Fehler und geben Sie die korrekte Post-Order Reihenfolge an. (Zur Erinnerung: Die *In-Order* Traversierung eines binären Suchbaums ist immer die sortierte Folge.)

1 10 7 5 17 14 12 26 30 28 25 15

9 11 10 8 13 17 14 21 20 18 16 12

1 4 3 11 16 14 15 13 19 20 17 5



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2014

30. April 2014

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Unterschrift:

Anzahl abgegebener Zusatzblätter:

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. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte!).

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

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

Viel Glück!

Aufgabe 1.B: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (8 Punkte)

Gegeben sei die folgende Funktion $f : \mathbb{N}^+ \rightarrow \mathbb{R}$

$$f(n) = \begin{cases} e^{-n} \cdot \sin^2(n) + \frac{1}{2 \ln(n)} + n & \text{wenn } n > 42 \text{ und gerade} \\ e \cdot \sin^2(n) + \frac{3}{4}n^2 - \ln(n) & \text{wenn } n \leq 42 \\ n^3 + e^{-n+1} \cdot \sin^2(n) + \ln(n) & \text{sonst} \end{cases}$$

Gehen Sie davon aus, dass das Argument von $\sin()$ in Grad angegeben wird.
 Kreuzen Sie **in der folgenden Tabelle** die zutreffenden Felder an:

$f(n)$ ist	$O(\cdot)$	$\Omega(\cdot)$	$\Theta(\cdot)$	keines
e^n				
2^n				
n^2				
n				

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

b) (8 Punkte)

Tragen Sie für die Codestücke A und B jeweils die **Laufzeit** in Abhängigkeit von n und die **Werte der Variablen a und b** nach dem Ausführen des Codes in Θ -Notation in die unten stehenden Tabellen ein.

A $a = 0;$
 $b = n^4;$
solange $n > 1$ {
 $a = a + 1;$
 $c = n;$
für $d = 1, \dots, \log_2(c)$
 {
 $n = \lfloor \frac{n}{2} \rfloor;$
 }
 $n = n - 1;$
 }

B $a = n^2;$
 $b = \log_2(n);$
wiederhole
 $b = 4 \cdot b;$
für $i = 1, \dots, \sqrt{n}$ {
 $a = a - 1;$
 }
bis $a \leq 1;$

Laufzeit	a	b
$\Theta(\quad)$	$\Theta(\quad)$	$\Theta(\quad)$

Laufzeit	a	b
$\Theta(\quad)$	$\Theta(\quad)$	$\Theta(\quad)$

Aufgabe 2.B: Sortierverfahren

(18 Punkte)

Gegeben sei der folgende Algorithmus **Sortiere**(A, n) in Pseudocode, der für ein Feld $A = (A[1], \dots, A[n])$ aufgerufen wird:

Sortiere(A, n)

wiederhole

vertauscht = *falsch*;

für $i = 1, 2, \dots, n - 1$ {

falls $A[i + 1] \geq A[i]$ **dann** {

vertausche $A[i + 1]$ und $A[i]$;

vertauscht = *wahr*;

}

}

bis *vertauscht* == *falsch*;

- (7 Punkte)

Überlegen Sie sich die Funktionsweise anhand der Eingabefolge

$$A = \langle 4, 3, 7, 6, 1, 2 \rangle$$

Wenden Sie **Sortiere**(A, n) auf diese Folge an und schreiben Sie dabei den Zustand der Eingabefolge zu Beginn jedes Durchlaufs der **wiederhole**-Schleife auf.

Beschreiben Sie anschließend mit wenigen Worten die **Funktionsweise**.

- (4 Punkte)

Algorithmus **Sortiere**(A, n) arbeitet nicht für jede beliebige Zahlenfolge korrekt. Beschreiben Sie in wenigen Worten, was genau das Problem ist und wie es behoben werden kann.

- (5 Punkte)

Bestimmen Sie die Laufzeit im Worst-Case der korrigierten Version von **Sortiere**(A, n) in Θ -Notation in Abhängigkeit der Eingabegröße n .

Geben Sie anschließend eine Folge mit vier Zahlen an, bei der die Worst-Case Laufzeit zustande kommt.

- (2 Punkte)

Ist der korrigierte Algorithmus stabil? Begründen Sie Ihre Antwort!

Aufgabe 3.B: Binäre Suchbäume

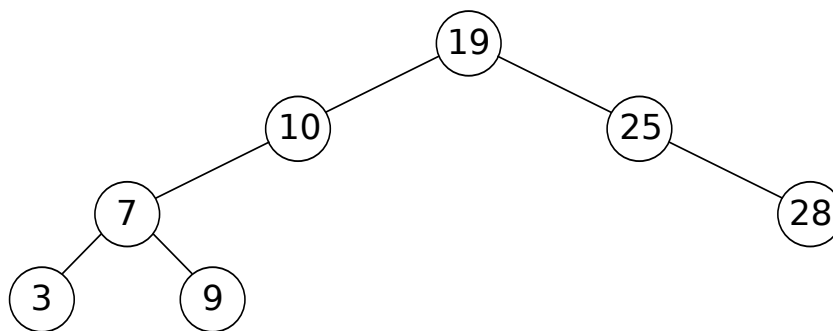
(16 Punkte)

a) (7 Punkte)

Gegeben ist folgender binärer Suchbaum.

- Fügen Sie die Elemente 15, 14 und 23 in den Baum ein.
- Löschen Sie anschließend die Elemente 15 und 19.
- Fügen Sie danach die Elemente 12 und 15 ein.

Falls Sie einen Ersatzknoten benötigen, verwenden Sie den Predecessor. Führen Sie alle Operationen in der angegebenen Reihenfolge aus und zeichnen Sie den Baum nach jeder einzelnen Einfüge- bzw. Löschoperation.



b) (9 Punkte)

Bei den folgenden *Post-Order* Traversierungen binärer Suchbäume ist jeweils ein Fehler unterlaufen. Finden Sie für jeden Baum den Fehler und geben Sie die korrekte Post-Order Reihenfolge an. (Zur Erinnerung: Die *In-Order* Traversierung eines binären Suchbaums ist immer die sortierte Folge.)

4 3 6 9 7 8 5 13 19 17 14 10

3 12 10 21 18 17 19 23 30 26 22 16

1 6 5 9 14 11 10 15 18 17 13 7