



186.172 Algorithmen und Datenstrukturen 1 VL 4.0

1. Übungstest SS 2010

30. April 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

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

Die Arbeitszeit beträgt 55 Minuten.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	16	17	17	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) (4 Punkte)

Sie wollen beweisen, dass für die folgende Funktion $f(n)$ die Beziehung $f(n) = \Theta(n^2)$ gilt. Geben Sie jeweils einen gültigen Wert für $c_1 \in \mathbb{N}$ und $c_2 \in \mathbb{N}$ an, wenn für n_0 der Wert 10 angenommen wird? *Hinweis:* Der eigentliche Beweis, sowie die formale Definition müssen nicht angeführt werden.

$$f(n) = \begin{cases} \frac{2 \cdot n^5}{20 \cdot n^3} + 4 \cdot n^2 - 5 \cdot n, & \text{falls } n \text{ ein Vielfaches von } 3 \text{ ist} \\ 10 + 20 \cdot n^2 - \frac{1}{n^4}, & \text{falls } n \text{ kein Vielfaches von } 3 \text{ ist} \end{cases}$$

$c_1 = \underline{\hspace{2cm}}$

$c_2 = \underline{\hspace{2cm}}$

b) (6 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Kreuzen Sie in der folgenden Tabelle jeweils alle richtigen Folgerungen an. Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

Annahme	Folgerung: $f(n)$ ist in			
	$\Omega(g(n))$	$O(g(n))$	$\Omega(h(n))$	$O(h(n))$
$(f(n) = O(2 \cdot g(n))) \wedge (h(n) = O(3 \cdot g(n)))$				
$(f(n) = \Omega(g(n) + h(n))) \wedge (h(n) = \Theta(f(n)))$				

c) (6 Punkte)

Bestimmen Sie die Laufzeiten der unten angegebenen Codestücke A und B in Abhängigkeit von n in Θ -Notation.

A

```

i = 1;
k = 1;
for j = 1, ..., n do
  while i < n do
    i = i + 1;
    k = k + i;
  end while
  i = 1;
end for
j = n;
for i = 1, ..., k do
  j = 2 · j + i;
end for
    
```

B

```

a = 1;
b = n;
for i = 1, ..., n do
  while b > 0 do
    b = b - √n;
    if b < √n then
      a = 2 · a;
    end if
  end while
end for
for k = 1, ..., a do
  b = 2 · b + a;
end for
    
```

Aufgabe 2.A: Sortierverfahren

(17 Punkte)

a) (6 Punkte)

Gegeben sind folgende binäre Zahlen:

$\langle 10, 11, 1, 0, 101, 111, 01, 00 \rangle$

Diese Zahlen sollen mit Hilfe des Verfahrens *Sortieren durch Fachverteilung* aufsteigend sortiert werden. Wählen Sie die Anzahl der Fächer entsprechend der Zahlenbasis und die Anzahl der Sammel- und Verteilungsphasen basierend auf der maximalen Stellenanzahl.

Geben Sie die Inhalte aller Fächer nach jeder Verteilungsphase und die Reihenfolge der Zahlen nach jeder Sammelphase an.

b) (11 Punkte)

Gegeben sei der Algorithmus $\text{sort}(A, i)$, wobei i ein Index im Feld A mit den zu sortierenden Elementen $A[1] \dots A[n]$ ist. Der Schlüsselwert eines Elementes $A[i]$ wird im Attribut $A[i].key$ gespeichert. Der Algorithmus wird durch den Aufruf $\text{sort}(A, n)$ gestartet.

Algorithmus: $\text{sort}(A, i)$

Eingabe: Feld A , Index i im Feld A

Rückgabewert: Sortiertes Feld A

```
if  $i > 1$  then
   $max = i$ ;
   $j = i - 1$ ;
  while  $j > 0$  do
    if  $A[j].key > A[max].key$  then
       $max = j$ ;
    end if
     $j = j - 1$ ;
  end while
   $maxelement = A[max]$ ;
   $A[max] = A[i]$ ;
   $A[i] = maxelement$ ;
   $i = i - 1$ ;
   $A = \text{sort}(A, i)$ ;
end if
return  $A$ ;
```

- Auf welchem aus der Vorlesung bekannten Sortierverfahren beruht $\text{sort}()$?
- Geben Sie die Laufzeit von $\text{sort}()$ im Best- und im Worst-Case in Θ -Notation in Abhängigkeit der Anzahl der zu sortierenden Element n an.
- Wenden Sie den Algorithmus $\text{sort}()$ auf die Zahlenfolge $\langle 5, 5, 1, 2, 1 \rangle$ an. Geben Sie die Zahlenfolge jeweils bei dem Aufruf von $\text{sort}()$ an.
- Wie müssten Sie den Algorithmus abändern, damit die Elemente der Felder in umgekehrter Reihenfolge sortiert werden?

Aufgabe 3.A: Abstrakte Datentypen und Suchverfahren

(17 Punkte)

- a) (13 Punkte) Schreiben Sie in detailliertem Pseudocode eine Funktion *checkBinarySearchTree*, die rekursiv überprüft, ob es sich um einen gültigen binären Suchbaum handelt. *root* ist der Wurzelknoten des zu überprüfenden binären Baumes. Ein Knoten *p* dieses Baumes enthält folgende Informationen:
- *p.left*: Zeiger auf sein linkes Kind, *null* falls kein linkes Kind existiert
 - *p.right*: Zeiger auf sein rechtes Kind, *null* falls kein rechtes Kind existiert
 - *p.key*: Schlüsselwert des Knotens

Geben Sie für Ihren Algorithmus den Aufwand für den Worst-Case in Θ -Notation in Abhängigkeit der Anzahl n der Schlüssel, die in dem binären Baum gespeichert sind, an.

- b) (4 Punkte)

Gegeben ist die Zahlenfolge

$\langle 33, 5, 15, 10, 12, 9, 4, 11 \rangle$.

Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren B-Baum der Ordnung 5 ein. Zeichnen Sie den resultierenden B-Baum. (die leeren Blätter können bei der Zeichnung entfallen).



186.172 Algorithmen und Datenstrukturen 1 VL 4.0

1. Übungstest SS 2010

30. April 2010

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

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

Die Arbeitszeit beträgt 55 Minuten.

	B1:	B2:	B3:	Summe:
Erreichbare Punkte:	17	16	17	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Glück!

Aufgabe 1.B: Abstrakte Datentypen und Suchverfahren

(17 Punkte)

a) (4 Punkte)

Gegeben ist die Zahlenfolge

$\langle 30, 2, 12, 7, 9, 6, 1, 8 \rangle$.

Fügen Sie alle Schlüssel der Folge nacheinander in einen anfangs leeren B-Baum der Ordnung 5 ein. Zeichnen Sie den resultierenden B-Baum. (die leeren Blätter können bei der Zeichnung entfallen).

b) (13 Punkte) Schreiben Sie in detailliertem Pseudocode eine Funktion *checkBinarySearchTree*, die rekursiv überprüft, ob es sich um einen gültigen binären Suchbaum handelt. *root* ist der Wurzelknoten des zu überprüfenden binären Baumes. Ein Knoten *p* dieses Baumes enthält folgende Informationen:

- *p.left*: Zeiger auf sein linkes Kind, *null* falls kein linkes Kind existiert
- *p.right*: Zeiger auf sein rechtes Kind, *null* falls kein rechtes Kind existiert
- *p.key*: Schlüsselwert des Knotens

Geben Sie für Ihren Algorithmus den Aufwand für den Worst-Case in Θ -Notation, in Abhängigkeit der Anzahl *n* der Schlüssel, die in dem binären Baum gespeichert sind, an.

Aufgabe 2.B: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (6 Punkte)

Bestimmen Sie die Laufzeiten der unten angegebenen Codestücke A und B in Abhängigkeit von n in Θ -Notation.

```
A
  y = n;
  x = 1;
  for i = 1, ..., n3 do
    while y > 0 do
      y = y - √n;
      if y < √n then
        x = 2 · x;
      end if
    end while
  end for
  for k = 1, ..., x do
    y = 2 · y + x;
  end for
```

```
B
  k = 1;
  a = 1;
  for b = 1, ..., n do
    while a < n do
      a = a + 1;
      k = k + a;
    end while
    a = 1;
  end for
  b = n;
  for a = 1, ..., k do
    b = 2 · b + a;
  end for
```

b) (4 Punkte)

Sie wollen beweisen, dass für die folgende Funktion $f(n)$ die Beziehung $f(n) = \Theta(n^3)$ gilt. Geben Sie jeweils einen gültigen Wert für $c_1 \in \mathbb{N}$ und $c_2 \in \mathbb{N}$ an, wenn für n_0 der Wert 5 angenommen wird? *Hinweis:* Der eigentliche Beweis, sowie die formale Definition müssen nicht angeführt werden.

$$f(n) = \begin{cases} \frac{2 \cdot n^6}{20 \cdot n^3} + 4 \cdot n^3 - 5 \cdot n, & \text{falls } n \text{ ein Vielfaches von } 5 \text{ ist} \\ 10 + 30 \cdot n^3 - \frac{1}{n^4}, & \text{falls } n \text{ kein Vielfaches von } 5 \text{ ist} \end{cases}$$

$c_1 = \underline{\hspace{2cm}}$

$c_2 = \underline{\hspace{2cm}}$

c) (6 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ Funktionen mit positivem Wertebereich. Kreuzen Sie in der folgenden Tabelle jeweils alle richtigen Folgerungen an. Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

Annahme	Folgerung: $f(n)$ ist in			
	$\Omega(h(n))$	$O(h(n))$	$\Omega(g(n))$	$O(g(n))$
$(f(n) = \Omega(g(n) + h(n))) \wedge (h(n) = \Theta(f(n)))$				
$(f(n) = O(2 \cdot g(n))) \wedge (h(n) = O(3 \cdot g(n)))$				

Aufgabe 3.B: Sortierverfahren

(17 Punkte)

a) (11 Punkte)

Gegeben sei der Algorithmus `sort(A, i)`, wobei i ein Index im Feld A mit den zu sortierenden Elementen $A[1] \dots A[n]$ ist. Der Schlüsselwert eines Elementes $A[i]$ wird im Attribut $A[i].key$ gespeichert. Der Algorithmus wird durch den Aufruf `sort(A, n)` gestartet.

Algorithmus: `sort(A, i)`

Eingabe: Feld A , Index i im Feld A

Rückgabewert: Sortiertes Feld A

```
if  $i > 1$  then
   $max = i$ ;
   $j = i - 1$ ;
  while  $j > 0$  do
    if  $A[j].key > A[max].key$  then
       $max = j$ ;
    end if
     $j = j - 1$ ;
  end while
   $maxelement = A[max]$ ;
   $A[max] = A[i]$ ;
   $A[i] = maxelement$ ;
   $i = i - 1$ ;
   $A = sort(A, i)$ ;
end if
return  $A$ ;
```

- Auf welchem aus der Vorlesung bekannten Sortierverfahren beruht `sort()`?
- Geben Sie die Laufzeit von `sort()` im Best- und im Worst-Case in Θ -Notation in Abhängigkeit der Anzahl der zu sortierenden Element n an.
- Wenden Sie den Algorithmus `sort()` auf die Zahlenfolge $\langle 7, 7, 3, 4, 3 \rangle$ an. Geben Sie die Zahlenfolge jeweils bei dem Aufruf von `sort()` an.
- Wie müssten Sie den Algorithmus abändern, damit die Elemente der Felder in umgekehrter Reihenfolge sortiert werden?

b) (6 Punkte)

Gegeben sind folgende binäre Zahlen:

$\langle 11, 10, 1, 0, 01, 111, 101, 00 \rangle$

Diese Zahlen sollen mit Hilfe des Verfahrens *Sortieren durch Fachverteilung* aufsteigend sortiert werden. Wählen Sie die Anzahl der Fächer entsprechend der Zahlenbasis und die Anzahl der Sammel- und Verteilungsphasen basierend auf der maximalen Stellenanzahl.

Geben Sie die Inhalte aller Fächer nach jeder Verteilungsphase und die Reihenfolge der Zahlen nach jeder Sammelphase an.