

# Präsenzübung

## Datenstrukturen und Algorithmen (SS 2013)

Prof. Dr. Leif Kobbelt  
Thomas Ströder, Fabian Emmes,  
Sven Middelberg, Michael Kremer

Dienstag, 28. Mai 2013

Nachname: \_\_\_\_\_

Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Studiengang: \_\_\_\_\_

- Schreiben Sie auf jedes Blatt **Vorname**, **Name** und **Matrikelnummer**.
- Geben Sie Ihre Antworten in lesbarer und verständlicher Form an.
- Bitte beantworten Sie die Aufgaben auf den **Aufgabenblättern**. Sollten Sie ggf. zusätzliches Papier benötigen, so wird Ihnen dies von uns ausgehändigt. Selbst mitgebrachtes Papier ist nicht zugelassen. Antworten auf anderen Blättern können nur berücksichtigt werden, wenn **Name, Matrikelnummer und Aufgabennummer** deutlich darauf erkennbar sind.
- Was nicht bewertet werden soll, kennzeichnen Sie bitte durch **Durchstreichen**.
- Werden Täuschungsversuche beobachtet, so wird die Präsenzübung mit **0 Punkten** bewertet.
- Um die Klausurzulassung zu erreichen, benötigen Sie **mindestens 33%** der hier erreichbaren Punkte.
- Verwenden Sie ausschließlich dokumentenechte Stifte mit schwarzer oder blauer Farbe. Insbesondere sind keine Bleistifte und keine Stifte mit roter Farbe zugelassen.

	Thema	Punkte	Erreichte Punkte
1	Allgemeine Fragen	20	
2	Binärbäume	20	
3	Sortierverfahren	20	
4	Laufzeitanalyse	20	
5	ADT	20	
	Summe	100	

<b>Vorname</b>	<b>Name</b>	<b>Matr.-Nr.</b>	<b>Seite</b>
			<b>1 / 9</b>

**Aufgabe 1** (*Allgemeine Fragen* [20 Punkte])

1. Tragen Sie in der folgenden Tabelle die *Best-*, *Average-* und *Worst-Case*-Komplexitäten der jeweiligen Sortierverfahren in  $O$ -Notation ein. [3 Punkte]

	<i>Best-Case-Kompl.</i>	<i>Avg.-Case-Kompl.</i>	<i>Worst-Case-Kompl.</i>
Bubble-Sort			
Selection-Sort			
Insertion-Sort			
Quick-Sort			
Merge-Sort			
Heap-Sort			

2. Gegeben sei eine nahezu vorsortierte Folge, von der Sie wissen, dass nur wenige Elemente unsortiert sind. Ihnen stehen vier Sortierverfahren zur Verfügung: **Bubble-Sort**, **Insertion-Sort**, **Quick-Sort** und **Merge-Sort**. Welches der vier Verfahren würden Sie zur Sortierung der Folge empfehlen? Begründen Sie Ihre Antwort in einem Satz. [3 Punkte]

3. Teilen Sie die folgenden Funktionen derart in Äquivalenzklassen auf, dass  $f$  und  $g$  genau dann in derselben Äquivalenzklasse sind, wenn  $f \in \Theta(g)$  gilt. [3 Punkte]

$$f_1(x) = x^2$$

$$f_2(x) = x \log x$$

$$f_3(x) = x^2 + x \log x$$

$$f_4(x) = x$$

$$f_5(x) = 2^x$$

$$f_6(x) = 1/x$$

$$f_7(x) = \sqrt{x}$$

$$f_8(x) = x(2 + \sin x)$$

$$f_9(x) = 1$$

Vorname	Name	Matr.-Nr.	Seite
			2 / 9

4. Sei  $f \in O(n^2 \log n)$  und  $g \in O(2^n)$ . Geben Sie eine *möglichst kleine* obere Schranke für die Funktion  $f \circ g$  in  $O$ -Notation an. [3 Punkte]

5. Ein Ternärbaum ist ein Baum, in dem jeder Knoten maximal drei Söhne hat. Wieviele Knoten hat ein Ternärbaum der Höhe  $h$  minimal und maximal? [3 Punkte]

6. Ist *dynamische Programmierung* ein *top-down*- oder ein *bottom-up*-Verfahren? [2 Punkte]

7. In der Vorlesung haben Sie verschiedene Entwurfsparadigmen kennengelernt. Nach welchem Entwurfsparadigma wurde der folgende Algorithmus entworfen? Sie müssen Ihre Antwort nicht begründen. [3 Punkte]

```

Sum(A[l..r])
  if (r == l) then
    return A[r]
  m ← ⌊ $\frac{l+r}{2}$ ⌋
  return Sum(A[l..m]) + Sum(A[m + 1..r])

```

Vorname	Name	Matr.-Nr.	Seite
			3 / 9

**Aufgabe 2** (*Binärbäume* [20 Punkte])

1. In der Vorlesung haben Sie die Methoden `ReadTerm`, `ReadProduct` und `ReadFactor` kennengelernt, die aus einem arithmetischen Term einen Binärbaum konstruieren. Geben Sie den Binärbaum an, welcher bei Aufruf der Methode `ReadTerm` für den Term

$$A * B / (C - D + E) + F - (G + H / (I - J) - K) * L$$

entsteht. [14 Punkte]

2. In der Vorlesung haben Sie weiterhin Algorithmen zur Traversierung von Binärbäumen kennengelernt, die sich darin unterscheiden, ob der zugrundeliegende arithmetische Term in *Präfix*-, *Infix*- oder *Postfixreihenfolge* traversiert wird. Geben Sie für alle drei Traversierungsmethoden an, in welcher Reihenfolge die Knoten des von Ihnen konstruierten Binärbaums aus Aufgabenteil 1 traversiert werden. [6 Punkte]





Vorname	Name	Matr.-Nr.	Seite
			6 / 9

**Aufgabe 4** (*Laufzeitanalyse/Rekursionsgleichungen* [20 Punkte])

1. Gegeben sei folgender Java-Code:

```
public static int f(int n) {  
    if (n == 1) {  
        return 3;  
    } else {  
        return f(0.25*n) + Math.sqrt(n) - 1;  
    }  
}
```

Überführen Sie die Funktion  $f$  in eine Rekursionsgleichung für die Laufzeitabschätzung unter der Annahme, dass der Eingabeparameter  $n$  stets positiv ist und alle verwendeten arithmetischen Operationen und Vergleiche konstante Laufzeit  $c$  haben. [6 Punkte]

2. Gegeben sei folgende Rekursionsgleichung:

$$T(1) = c$$
$$T(n) = T(\lfloor n/2 \rfloor) + 2n + c,$$

wobei  $c \in \mathbb{R}$  eine Konstante ist. Bestimmen Sie eine möglichst kleine, nicht-rekursive obere Schranke für  $T(n)$  ohne Zuhilfenahme des Master-Theorems und ohne Verwendung der  $O$ -Notation. **Hinweis:** Es gilt  $\sum_{i=0}^n q^i = \frac{1-q^{n+1}}{1-q}$ . [10 Punkte]

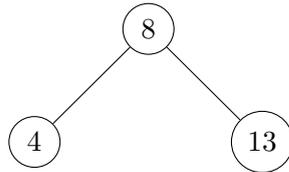
Vorname	Name	Matr.-Nr.	Seite
			7 / 9

3. Benutzen Sie nun das **Master-Theorem** zur Bestimmung der asymptotischen Laufzeit der in Teilaufgabe 2 gegebenen Rekursionsgleichung. [4 Punkte]

Vorname	Name	Matr.-Nr.	Seite
			8 / 9

**Aufgabe 5** (*Abstrakte Datentypen* [20 Punkte])

Entwerfen Sie einen abstrakten Datentyp Set, der Mengen von natürlichen Zahlen als binäre Suchbäume darstellt. Beispielsweise entspricht der folgende binäre Suchbaum der Menge  $\{4, 8, 13\}$ :



Der ADT soll insbesondere aus den folgenden Funktionen bestehen:

create erzeugt eine leere Menge.  
 node erzeugt einen Knoten des Binärbaums.  
 insert fügt eine Zahl in eine Menge ein.  
 union erzeugt die Vereinigung von zwei Mengen.

Der oben abgebildete Baum würde zum Beispiel dem folgenden Term entsprechen:

$\text{node}(\text{node}(\text{create}(), 4, \text{create}()), 8, \text{node}(\text{create}(), 13, \text{create}()))$ .

1. Geben Sie geeignete Sorten (Datentypen) und Signaturen für die Funktionen create, node, insert und union an. [5 Punkte]

<b>Vorname</b>	<b>Name</b>	<b>Matr.-Nr.</b>	<b>Seite</b>
			<b>9 / 9</b>

2. Geben Sie geeignete Axiome an, die die Funktionen insert und union definieren. Beachten Sie, dass eine Menge keine doppelten Elemente enthalten darf. Der Vergleich von natürlichen Zahlen kann als elementare Operation betrachtet werden. [15 Punkte]