

Holger Füßler

A5, 6, Raum B 219
68131 Mannheim
Telefon: (0621) 181-2605
Email: fuessler@informatik.uni-mannheim.de

Robert Schiele

B6, 29, Raum C0.04
68131 Mannheim
Telefon: (0621) 181-2214
Email: rschiele@uni-mannheim.de

Praktische Informatik I
Wintersemester 2005/2006

8. Übungsblatt
Abgabe: 11. Januar 2006

Allgemeine Bearbeitungshinweise

Verwenden Sie für die Programmieraufgaben die in den Aufgaben angegebenen Klassennamen und geben Sie den Klassen-Quellcode sowohl auf Papier als auch per e-mail bei Ihrem Tutor ab. Falls nicht anders gefordert, lagern Sie sinnvoll Funktionalität in Methoden aus. Verwenden Sie Javadoc-Kommentare zur Dokumentation.

Aufgabe 1	8 Punkte
-----------	----------

Aufgabe 1 a)	5 Punkte
--------------	----------

Wer lügt und wer sagt die Wahrheit?

- Dieter sagt: Erika und Fred lügen.
- Erika sagt: wenn Dieter die Wahrheit sagt, dann lügt Gerd.
- Fred sagt: wenn Gerd die Wahrheit sagt, dann sagt auch Erika die Wahrheit.
- Gerd sagt: wenn Erika und Fred lügen, dann sagt Dieter die Wahrheit.

Stellen Sie aussagenlogische Gleichungen auf und finden Sie eine Lösung. Gibt es nur eine Lösung?

Aufgabe 1 b)	3 Punkte
--------------	----------

Vereinfachen Sie die folgende Formel durch Umformung unter Angabe der verwendeten Umformungsregeln.

$$((P \vee \neg Q) \rightarrow (P \wedge Q)) \leftrightarrow (R \rightarrow Q)$$

Aufgabe 2 a)

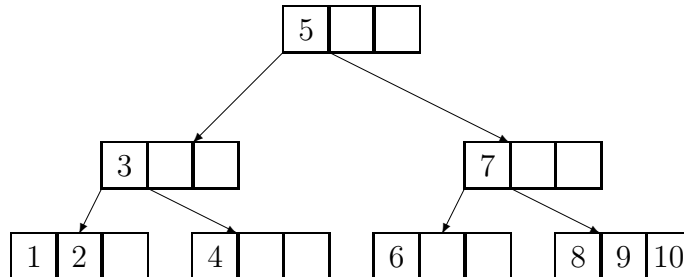
6 Punkte

Keine Programmieraufgabe! Fügen Sie die Werte 19, 7, 16, 11, 5, 8, 12, 2, 4, 10, 18 und 13 in dieser Reihenfolge in einen anfangs leeren B-Baum vom Rang $m = 4$ ein. Geben Sie den Baum nach jedem Schritt, bei dem sich dessen Struktur ändert, und am Ende an.

Aufgabe 2 b)

6 Punkte

Betrachten Sie nun folgenden B-Baum vom Rang $m = 4$.



Löschen Sie aus diesem Baum alle ungeraden Zahlen in aufsteigender Reihenfolge. Geben Sie den Baum nach jedem Schritt, bei dem sich dessen Struktur ändert, und am Ende an.

Aufgabe 2 c)

3 Punkte

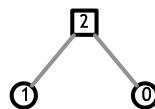
Ein Algorithmus zum Löschen von beliebigen Knoten aus einem B-Baum ist Ihnen bekannt. Auf welche Schwierigkeit stoßen Sie, wenn Sie einen Algorithmus zum Löschen eines beliebigen Knotens aus einem *binären* Suchbaum entwickeln wollen?

Gegeben Sei die folgende rekursive Implementierung der Fibonacci-Reihe.

```

1 public static long fiboRek(long n) {
    return (n < 2) ? n : ( fiboRek(n - 1) + fiboRek(n - 2) );
}
  
```

Ein Rekursionsbaum ist ein Baum, der die Berechnungsbeziehung zwischen den Elementen herstellt, also z.B. für $n = 2$



Aufgabe 3 a)

2 Punkte

Zeichnen Sie einen Rekursionsbaum der o.a. Methode für $n = 5$.

Aufgabe 3 b)

1 Punkte

Eine in der Datenverarbeitung oft verwendete Methode ist das Zwischenspeichern (Caching) von Ergebnissen. Im vorliegenden Fall könnte man jedes bereits berechnete Reihenglied zwischenspeichern und damit den Gesamtberechnungsaufwand verkleinern. Wie viele Methodenaufrufe kann man für $n = 5$ dadurch einsparen? Spielt es hierbei eine Rolle, ob man zuerst in den $n - 2$ Ast steigt? Begründen Sie mit dem Baum.

Aufgabe 3 c)

5 Punkte

Implementieren Sie eine Klasse `FiboCache`, in der die o.a. rekursive Fibonacci-Methode um Caching erweitert wird (Methodenname `fiborekCached()`). Hierbei dürfen Sie annehmen, dass n nicht größer als 1000 wird.

Aufgabe 3 d)

2 Punkte

Schreiben Sie eine main-methode, die die beiden Methoden `fiborek()` und `fiborekCached` für $n = 1, \dots, 40$ vergleicht. Geben Sie für jedes n und jede der Methoden aus (i) wie oft die Methode aufgerufen wurde, (ii) wie lange die Berechnung für dieses n gedauert hat. Achten Sie darauf, ihren cache zwischen den Berechnungen zurückzusetzen. (Benutzen Sie wie für das letzte Blatt Methode `System.nanoTime()`, um die Bearbeitungsdauer zu messen.)

Aufgabe 3 e)

1 Punkte

Angenommen, Sie würden den Cache zwischen den Aufrufen nicht löschen. Wie viele Aufrufe von `fiborekCached()` würden für die erstmalige Berechnung von n entstehen, wenn $n - 1$ bereits berechnet wurde und wieviele für jede weitere Berechnung von n ?

Aufgabe 3 f)

1 Punkte

Diskutieren Sie Vor- und Nachteile der gecachten rekursiven Methode gegenüber einem ungecachtem iterativen Ansatz. Für welchen Zweck würden Sie welche Implementierung einsetzen?