

2. Testklausur — Sommersemester 2013

Datenstrukturen und Algorithmen

24. Juni 2013

Bevor Sie mit der Bearbeitung dieser Klausur beginnen, lesen Sie bitte folgende Hinweise. Diese Hinweise sind bei der Bearbeitung zu beachten.

1. Prüfen Sie die Vollständigkeit Ihres Exemplars. Es sollte
 - dieses Hinweisblatt und
 - fünf Aufgaben auf fünf Blätternumfassen.
2. Tragen Sie auf jedem Lösungsblatt oben an den vorgesehenen Stellen Ihren Namen und Ihre Matrikelnummer ein. Blätter ohne diese Angaben werden nicht bewertet.
Hinter den Aufgaben ist jeweils hinreichend Platz für die Lösungen freigelassen. Reicht der Platz nicht aus, benutzen Sie die Rückseiten, wobei die Zuordnungen von Lösungen zu Aufgaben deutlich erkennbar sein müssen.
3. Geben Sie dieses Deckblatt zusammen mit den Aufgabenstellungen und den Lösungen sowie alles weitere beschriebene Papier ab.
4. Zugelassene Hilfsmittel: keine
5. Mit ≥ 30 Punkten haben Sie die Klausur bestanden.
6. Ergebnis (bitte nichts eintragen):

1 (15)	2 (10)	3 (10)	4 (15)	5 (10)	\sum_{Punkte} (60)	Note

Viel Erfolg!

Name:

Matrikel:

Aufgabe 1 (3+6+6=15 Punkte)

Gegeben sei der folgende Klassenrahmen zur Repräsentation von Suchbäumen:

```
public class SearchTree<T extends Comparable<T>> {  
  
    private class Node {        // Knoten im Suchbaum  
        T    value;            // Wert am Knoten  
        Node left;            // Wurzel des linken Unterbaums  
        Node right;           // Wurzel des rechten Unterbaums  
    }  
  
    private Node root;        // Wurzelknoten des gesamten Baums  
  
    ...                        // Definition der Methoden  
}
```

- (a) Schreiben Sie eine Methode, die die in einem Suchbaum enthaltenen Objekte in aufsteigend sortierter Reihenfolge ausgibt.

Hinweis: Natürlich dürfen Sie auch Hilfsmethoden definieren.

- (b) Schreiben Sie eine Methode `min()`, die das kleinste im Suchbaum enthaltene Objekt ermittelt. Formulieren Sie diese Methode iterativ (ohne Rekursion).
- (c) Lösen Sie die Fragestellung von Aufgabe (b) mit Hilfe einer rekursiven Methode.

Name:

Matrikel:

Aufgabe 2 (3+4+3=10 Punkte)

- (a) Geben Sie für die Schlüssel 11, 22, 35, 44, 52, 68, 77 einen Suchbaum mit maximaler und mit minimaler Höhe an.
- (b) In einen anfangs leeren Suchbaum werden nacheinander die folgenden Schlüssel eingefügt: 35, 28, 52, 7, 19, 65, 59, 44
Wie sieht der Baum nach dem Einfügen aller Schlüssel aus?
- (c) Wir löschen die 52. Wie sieht der Baum jetzt aus? Was passiert, wenn wir als nächstes die 35 löschen?

Name:

Matrikel:

Aufgabe 3 (5+5=10 Punkte)

(a) In einen leeren AVL-Baum werden nacheinander die folgenden Schlüssel eingefügt: 54, 32, 15, 8, 10, 67, 23, 13, 27

Wie sieht der AVL-Baum nach dem Einfügen von 15, 10, bzw. 27 aus?

(b) Wir löschen die 27 und anschließend die 67. Wie sieht der Baum jetzt aus?

Name:

Matrikel:

Aufgabe 4 (5+5+5=15 Punkte)

Gegeben ist die folgende Zahlenfolge:

45, 17, 29, 36, 18, 34, 52, 67, 39, 8, 12, 88, 55, 32, 61

Erläutern Sie schrittweise (siehe (a) bis (c)), wie diese Zahlenfolge mit Hilfe des Heap-Sort-Algorithmus aufsteigend sortiert wird.

- (a) Wie wird die Folge als Binärbaum repräsentiert und wie wird hieraus ein Binärbaum erzeugt, der die Heap-Eigenschaft erfüllt?
- (b) Geben Sie den aus der Folge konstruierten Heap an.
- (c) Führen Sie ausgehend von dem Heap in (b) drei Iterationen aus und geben Sie jeweils den dabei entstehenden Heap an.

Name:

Matrikel:

Aufgabe 5 (7+3=10 Punkte)

Gegeben Sei eine (zunächst leere) Hash-Tabelle der Größe 7, sowie die Hashfunktion

$$h(x) = x \bmod 7$$

- (a) Fügen Sie die folgenden Zahlen in die Hash-Tabelle ein, wobei Kollisionen durch Lineares Sondieren behandelt werden:

11, 22, 34, 7, 15

- (b) Welche Nachteile hat das Lineare Sondieren? Geben Sie Alternativen an.