

## Lösungen zur 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  $\geq 30$  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;            // Objekt 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 im Suchbaum enthaltenen Objekte in aufsteigend sortierter Reihenfolge ausgibt.
- (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.

### **Lösung:**

```
(a) public void printInInorder() {  
    printInInorder(root);  
}  
  
private void printInInorder(Node node) {  
    if (node==null) {  
        return;  
    }  
    printInInorder(node.left);  
    System.out.println(node.value);  
    printInInorder(node.right);  
}
```

**Name:**

**Matrikel:**

```
(b) public T min() {
    Node node;

    if (root == null) {
        return null;
    }

    node = root;
    while (node.left != null) {
        node = node.left;
    }

    return node.value;
}

(c) public T min() {
    return root!=null ? min(root) : null;
}

private T min(Node node) {
    return node.left==null ? node.value : min(node.left);
}
```

Name:

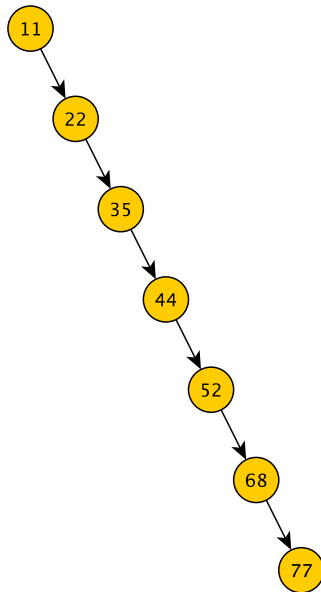
Matrikel:

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

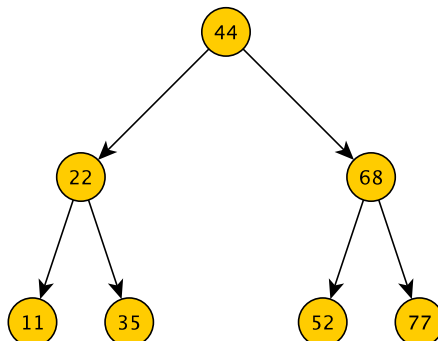
- (a) Geben Sie für die Schlüssel 11, 22, 35, 44, 52, 68, 77 einen Suchbaum mit maximaler und mit minimaler Höhe an.
- (b) In einen anfangs leeren Suchbaum werden nacheinander die folgenden Schlüssel eingefügt: 35, 28, 52, 7, 19, 65, 59, 44  
Wie sieht der Baum nach dem Einfügen aller Schlüssel aus?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

### Lösung:

- (a) Maximale Höhe:



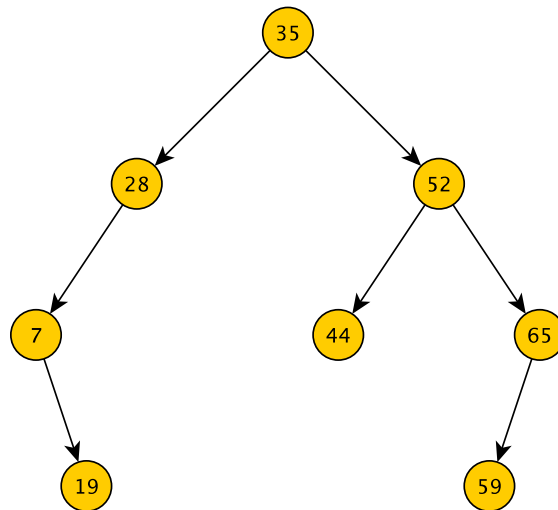
Minimale Höhe:



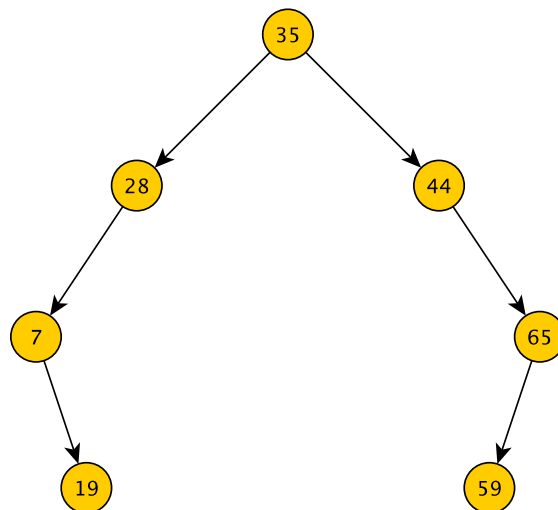
Name:

Matrikel:

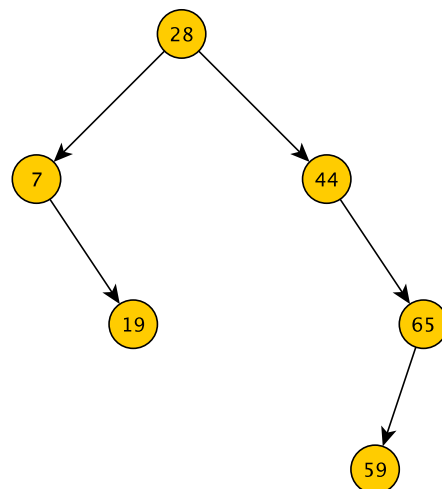
(b)



(c) Nach dem Löschen der 52:



Nach dem Löschen der 35:



Name:

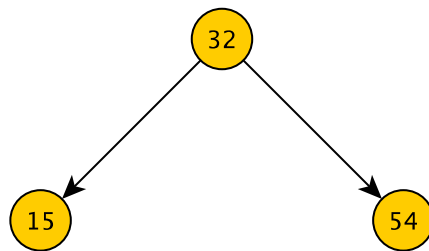
Matrikel:

### Aufgabe 3 (5+5=10 Punkte)

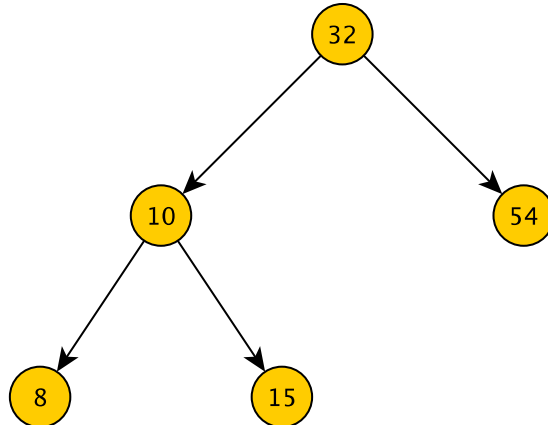
- (a) In einen leeren AVL-Baum werden nacheinander die folgenden Schlüssel eingefügt: 54, 32, 15, 8, 10, 67, 23, 13, 27  
Wie sieht der AVL-Baum nach dem Einfügen von 15, 10, bzw. 27 aus?
- (b) Wir löschen die 27 und anschließend die 67. Wie sieht der Baum jetzt aus?

### Lösung:

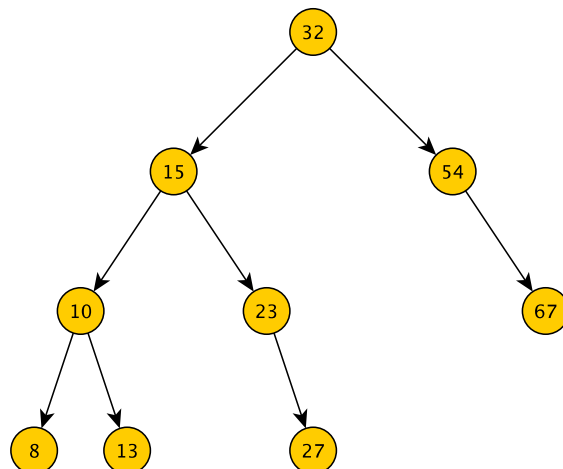
- (a) Nach dem Einfügen der 15:



Nach dem Einfügen der 10:



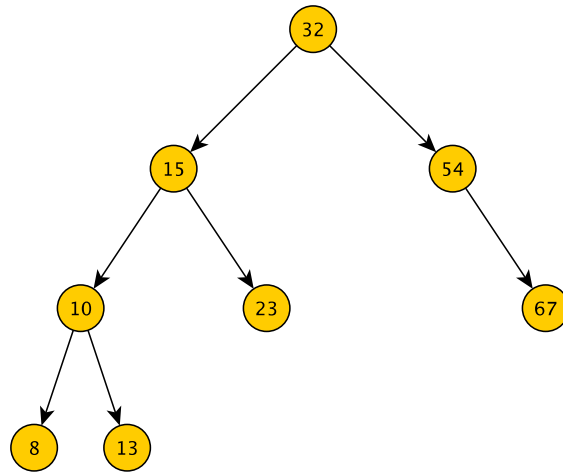
Nach dem Einfügen der 17:



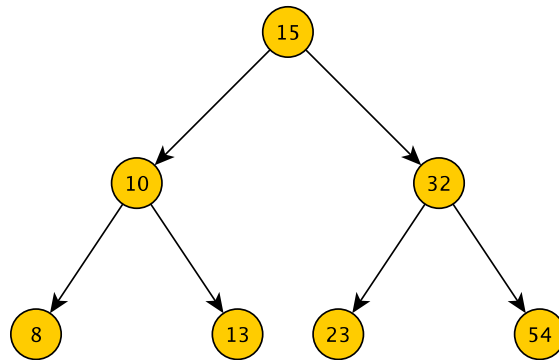
**Name:**

**Matrikel:**

(b) Nach dem Löschen der 27:



Nach dem Löschen der 67:



Name:

Matrikel:

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

Gegeben ist die folgende Zahlenfolge:

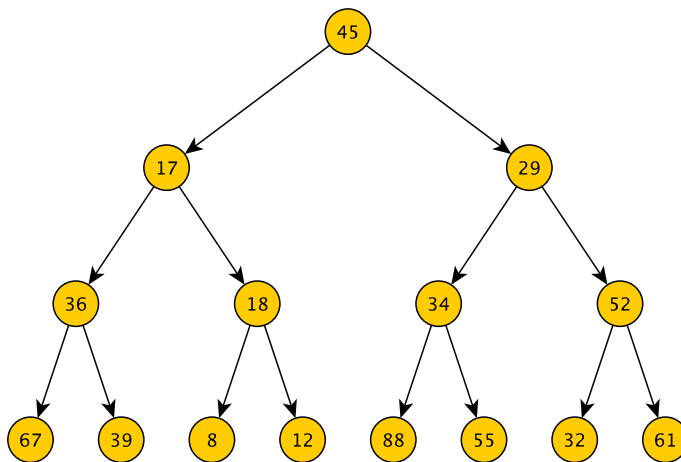
45, 17, 29, 36, 18, 34, 52, 67, 39, 8, 12, 88, 55, 32, 61

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

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

### Lösung:

- Zunächst wird ein linksvollständiger Baum aufgebaut. Da für die Schlüsselanzahl  $15 = 2^4 - 1$  gilt, ist der Baum sogar vollständig und hat die Höhe 4. In dem Baum werden die Schlüssel von oben nach unten und in jeder Ebene von rechts nach links platziert.



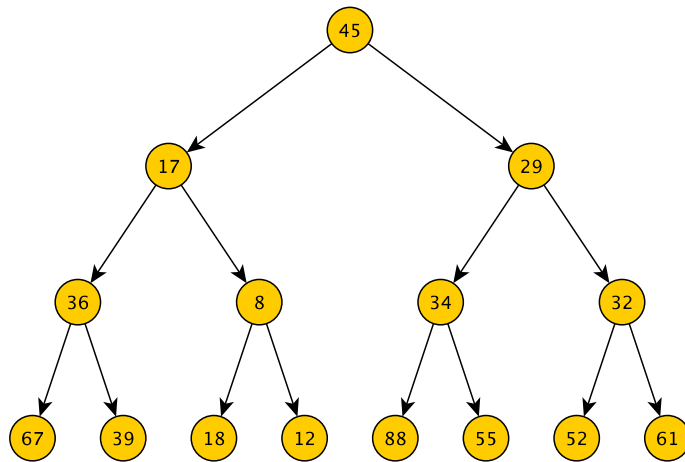
Nun wird von unten nach oben und in jeder Ebene von rechts nach links geprüft, ob die Heapbedingung erfüllt ist. Sollte sie an einem Knoten verletzt sein, findet ein Austausch mit dem kleineren der Söhne statt. Sollte nach dem Austausch am Sohn die Heapbedingung verletzt sein, wird am Sohn das Verfahren fortgesetzt, falls notwendig bis zur untersten Ebene.

Auf der zweituntersten Ebene finden folgende Vertauschungen statt: 52 mit 32 und 18 mit 8.

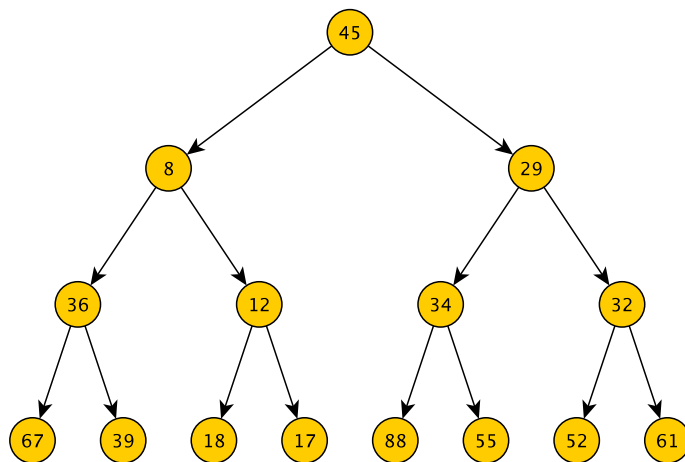


Name:

Matrikel:

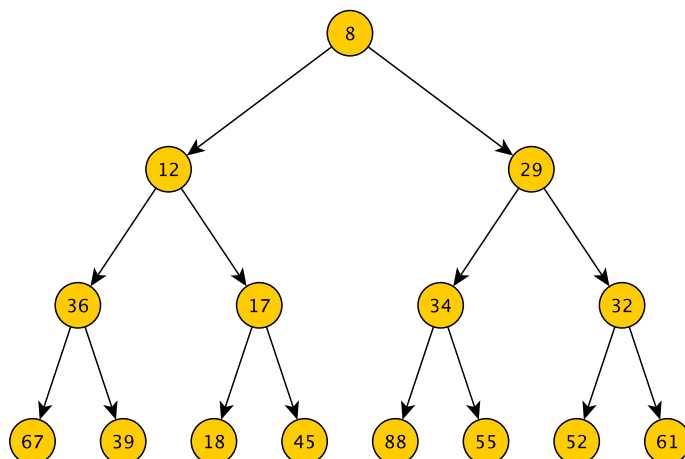


Auf der zweitobersten Ebene: 17 mit 8 und als Folge davon 17 mit 12.



Am Wurzelknoten: 45 mit 8 und als Folge davon 45 mit 12 und 45 mit 17.

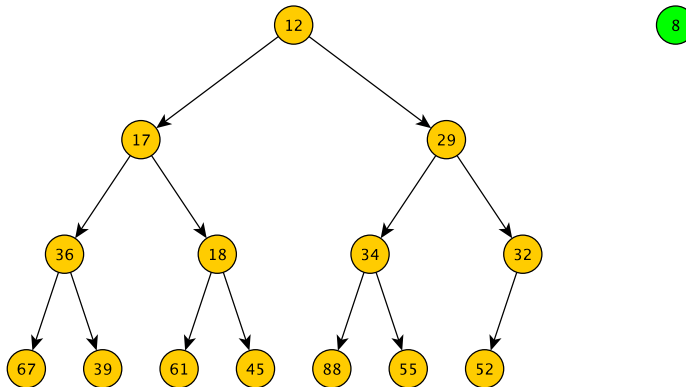
(b)



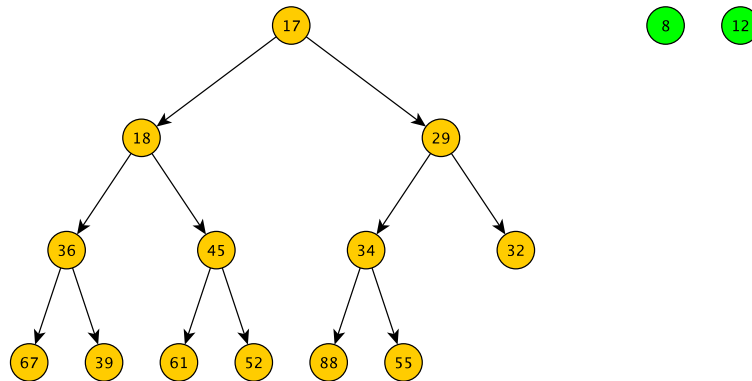
(c) Die 8 wird in die sortierte Folge übernommen und die 61 in die Wurzel platziert. Anschließend wird die 61 mit der 12, 17 und 18 getauscht.

Name:

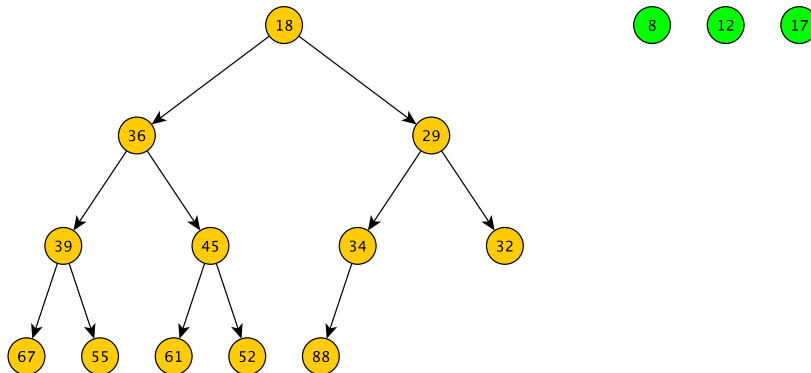
Matrikel:



Nun wird die 12 in die sortierte Folge übernommen und die 52 in die Wurzel platziert. Anschließend wird die 52 mit der 17, 18 und 45 getauscht.



Jetzt wird die 17 in die sortierte Folge übernommen und die 55 in der Wurzel platziert. Anschließend wird die 55 mit der 18, 36 und 39 getauscht.



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.

**Lösung:**

(a)

|   | leer | insert 11 | insert 22 | insert 34 | insert 7 | insert 15 |
|---|------|-----------|-----------|-----------|----------|-----------|
| 0 |      |           |           |           | 7        | 7         |
| 1 |      |           | 22        | 22        | 22       | 22        |
| 2 |      |           |           |           |          | 15        |
| 3 |      |           |           |           |          |           |
| 4 |      | 11        | 11        | 11        | 11       | 11        |
| 5 |      |           |           |           |          |           |
| 6 |      |           |           | 34        | 34       | 34        |

- (b) Ballungen an schon eingefügten Objekten.

Alternativen: Quadratisches Sondieren, Verkettung statt Sondieren