

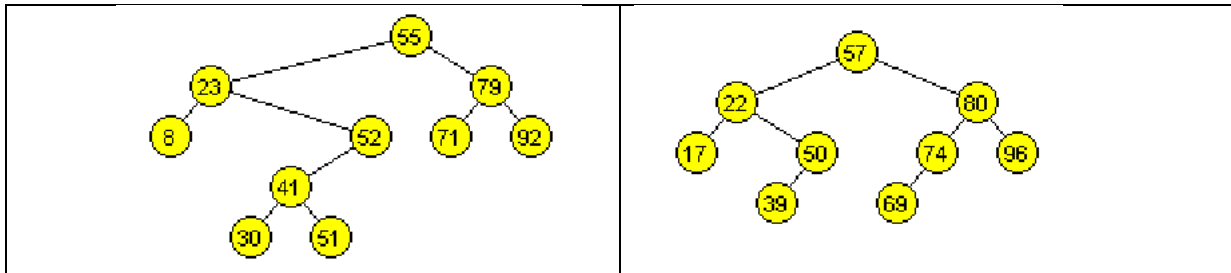
## Hausaufgaben 4

**20./21.04.2020**

Abgabe der Lösung am 26.04.2020

### Aufgabe 1

Schreiben Sie an die Knoten der beiden folgenden binären Suchbäume den AVL-Index und geben Sie anschließend an, ob es sich um AVL-Bäume handelt.



### Aufgabe 2

In Java gibt es verschiedene Klassen und Interfaces für assoziative Felder, die das Interface Map implementieren. Es gibt die Klassen HashMap, LinkedHashMap und TreeMap, sowie die Interfaces NavigableMap und SortedMap (die weiteren Klasse und Interfaces haben ganz spezielle Einsatzzwecke).

HashMap benutzt eine Hashtabelle, TreeMap benutzt einen Rot-Schwarz-Baum (spätere Vorlesung, ist aber ähnlich wie ein AVL-Baum).

- Beschreiben Sie die Datenstruktur der Klasse LinkedHashMap (-> API). Welcher Nachteil der Klasse HashMap wird mit der LinkedHashMap ausgeglichen?
- Was kann eine TreeMap mehr als eine HashMap? Beachten Sie, dass die TreeMap zusätzlich das Interface NavigableMap implementiert.

### Aufgabe 3

AVL-Suchbäume lassen sich auch als Listen verwenden. Die Grundidee ist, dass man in jedem Knoten die Anzahl der Knoten im linken und rechten Teilbaum speichert. Eine entsprechende Klasse gibt es in der Javastandardbibliothek leider nicht. Es gibt aber zwei sehr bekannte externe Bibliotheken: Apache Commons und Guava.

- Sehen Sie sich das Video TreeList.mp4 (ILIAS, Woche 4) an.
- Apache Commons und Guava stehen unter der Apache Commons Lizenz. Es ist wichtig, einen groben Überblick über die freien Softwarelizenzen zu haben. Lesen Sie dazu die folgende Seite:  
<https://exygy.com/blog/which-license-should-i-use-mit-vs-apache-vs-gpl/>  
Anmerkung: Dieser Punkt ist nicht klausurrelevant. Sie sollten die Seite in eigenem Interesse trotzdem lesen.
- Lassen Sie das Testprogramm ArrayListTest.java (Ilias, Woche 4) laufen und notieren Sie die Zeit.
- Ersetzen Sie die ArrayList durch eine TreeList.
- Notieren Sie erneut die Zeit.
- Schreiben Sie beide Zeiten und die jeweilige O-Klasse für das Einfügen eines Elements (Average Case) in eine Tabelle.

**Aufgabe 4**

Sehen Sie sich die folgende Seite mit den Kurzbeschreibungen einiger Algorithmen an:

<https://github.com/gibsjose/cpp-cheat-sheet/blob/master/Data%20Structures%20and%20Algorithms.md>

Fertigen Sie eine solche Kurzbeschreibung für den B-Baum an. Benutzen Sie folgende Einteilung:

**B-Baum**

Idee: Eigenschaften eines B-Baums, Ordnung eines B-Baums

Einfügen: Beschreibung des Einfügens

Löschen: Beschreibung des Löschens

O-Klassen: O-Klassen für Suchen/Einfügen/Löschen (Average+Worst Case)

**Aufgabe 5**

Zeichnen Sie jeweils die entstehenden B-Bäume.

a)		Löschen der 15
b)		Einfügen der 75
c)		Einfügen der 68

**Aufgabe 6**

Zeichnen Sie jeweils die entstehenden B-Bäume.

d)		Löschen der 43
e)		Löschen der 36
f)		Löschen der 90