5. Datenstrukturen und Algorithmen: Binäre Suchbäume und AVL-Bäume

(a) Geben Sie jeweils eine Definition für binäre Suchbäume und AVL-Bäume an.

Lösungsvorschlag

Binärer Suchbaum Für jeden Knoten gilt: Ein Knoten enthält einen Schlüsselwert. Ein Knoten hat zwei Kindknoten: einen linken und einen rechten Kindknoten. Alle Schlüsselwerte des linken Teilbaums sind kleiner, alle Schlüsselwerte des rechten Teilbaums sind größer als der Wert des Knotens.

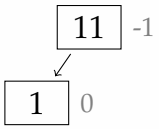
AVL-Baum Alle Eigenschaften des oben definierten binären Suchbaums gelten auch im AVL-Baum. Zusätzlich gilt: Die Differenz der Höhen des rechten und linken Teilbaums darf sich nie mehr als um eins unterscheiden.

(b) Zeichnen Sie einen AVL-Baum, der die folgenden Schlüssel enthält: 11, 1, 5, 37, 17, 29, 31, 3.

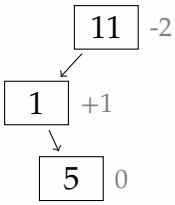
Lösungsvorschlag

Nach dem Einfügen von „11“:

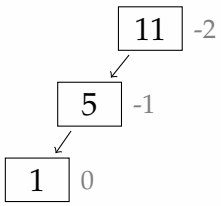
Nach dem Einfügen von „1“:



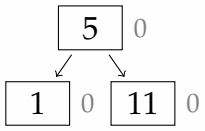
Nach dem Einfügen von „5“:



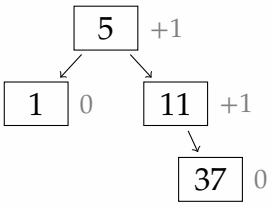
Nach der Linksrotation:



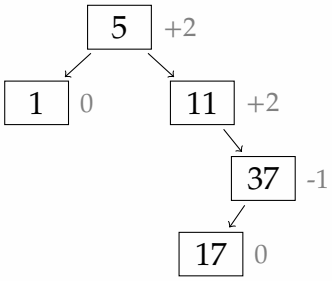
Nach der Rechtsrotation:



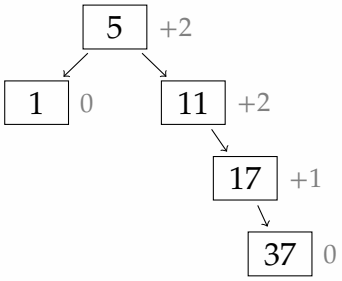
Nach dem Einfügen von „37“:



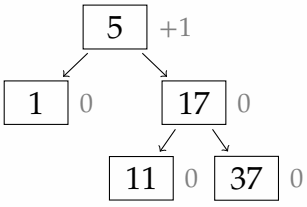
Nach dem Einfügen von „17“:



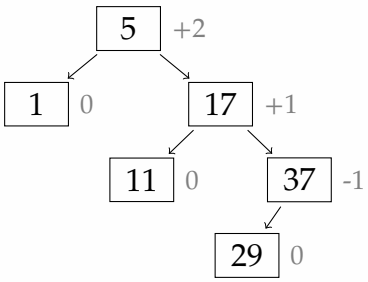
Nach der Rechtsrotation:



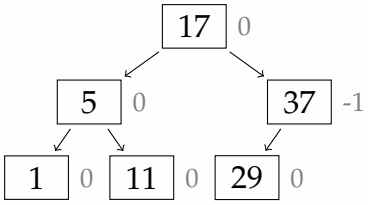
Nach der Linksrotation:



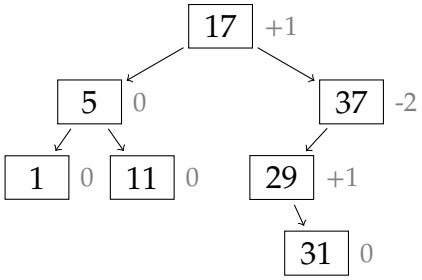
Nach dem Einfügen von „29“:



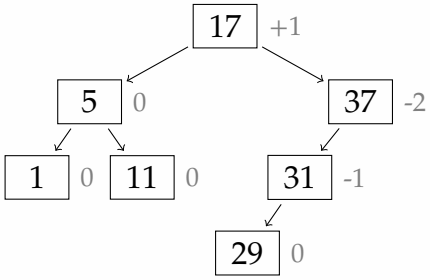
Nach der Linksrotation:



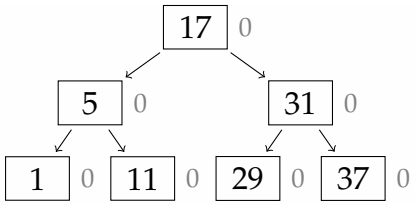
Nach dem Einfügen von „31“:



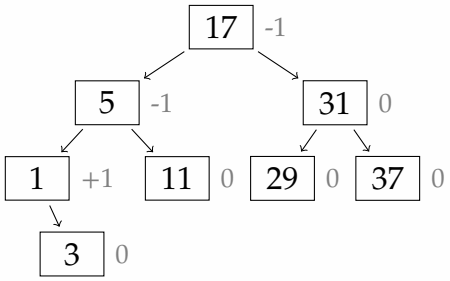
Nach der Linksrotation:



Nach der Rechtsrotation:



Nach dem Einfügen von „3“:



(c) Welche Zeitkomplexität haben die Operationen *Einfügen*, *Löschen* und *Suchen* auf AVL-Bäumen? Begründen Sie Ihre Antwort.

Lösungsvorschlag

Für alle Operationen wird jeweils nur ein Pfad von der Wurzel bis zum gewünschten Knoten beschritten. Für alle Operation ist deshalb die maximale Höhe des Baums entscheidend. Da AVL-Bäume höhenbalanciert sind.

(d) Implementieren Sie die Datenstruktur AVL-Baum mit Schlüsseln vom Typ int (für Java oder entsprechende Datentypen für die anderen Programmiersprachen) in entweder Java, C++, Smalltalk oder Eiffel. Ihre Implementierung muss als einzige Operation die Methode suche bereitstellen, die einen Schlüssel als Parameter bekommt und

- true zurückliefert, wenn der gesuchte Schlüssel im Baum enthalten ist,
- ansonsten false.

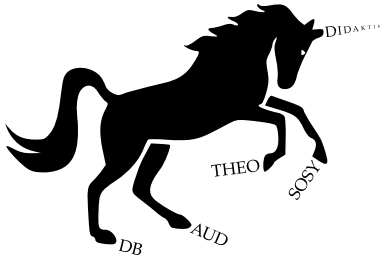
Ihre Implementierung muss keine Konstruktoren oder andere Methoden zur Initialisierung bereitstellen. Desweiteren soll die Implementierung nur Schlüsselknoten berücksichtigen.

Kommentieren Sie Ihre Implementierung ausführlich. Geben Sie an, welche Programmiersprache Sie gewählt haben.

```
public class AVLBaum {

}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46115/jahr_2010/fruehjahr/AVLBaum.java`



Die Bschlangaul-Sammlung

Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: