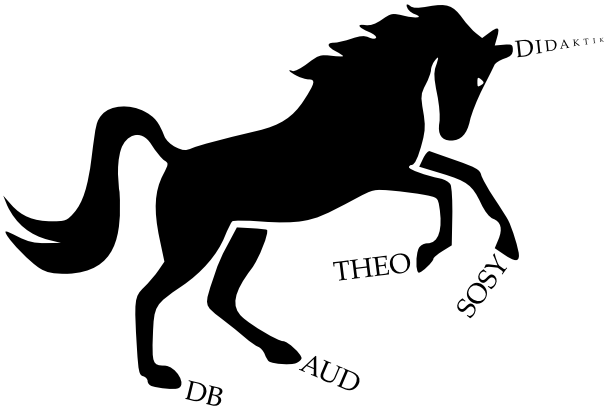
# 46115
# Frühjahr
# 2010

Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen

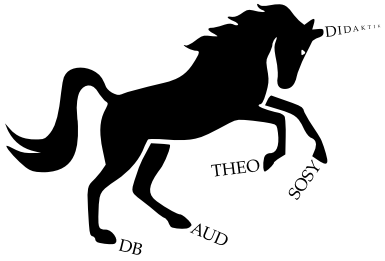



**Die Bschlangaul-Sammlung**

Hermine Bschlangaul and Friends

# Aufgabenübersicht






**Die Bschlangaul-Sammlung**
Hermine Bschlangaul and Friends

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

# Thema Nr. 1

**5. Datenstrukturen und Algorithmen: Binäre Suchbäume und AVL-Bäume [Binäre Suchbäume, AVL-Bäume, Implementierung]**

# 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.

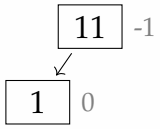
**AVL-Baum** Alle Eigenschaften des oben definierten binären Suchbaums gelten auch im AVL-Baum. Zusätzlich gilt: Die Differenz der Höhen des rechten und linken Teilbaums darf sich nie mehr als um eins unterscheiden.

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

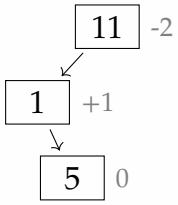
Lösungsvorschlag

*Nach dem Einfügen von „11“*:

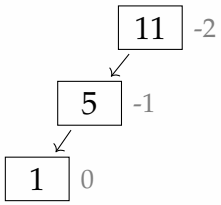

*Nach dem Einfügen von „1“*:



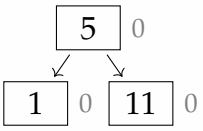

*Nach dem Einfügen von „5“*:



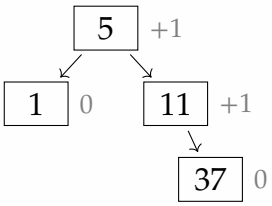
*Nach der Linksrotation*:



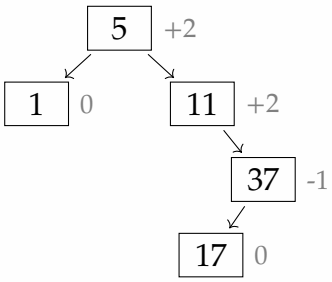

*Nach der Rechtsrotation*:



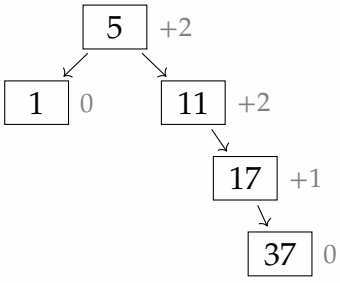

*Nach dem Einfügen von „37“*:



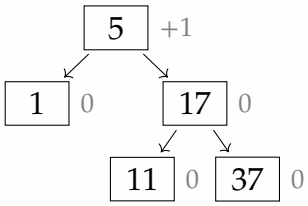

*Nach dem Einfügen von „17“*:



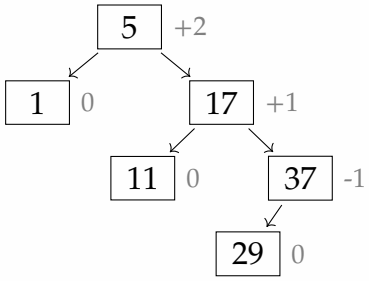

*Nach der Rechtsrotation*:



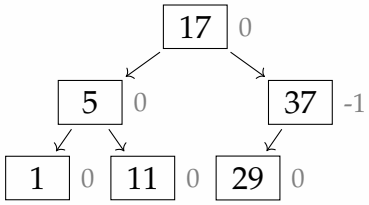
*Nach der Linksrotation*:



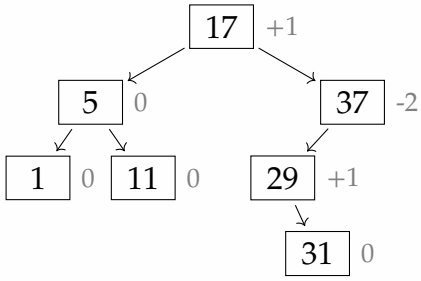

*Nach dem Einfügen von „29“*:



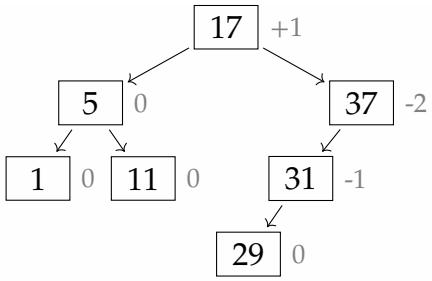

*Nach der Linksrotation*:



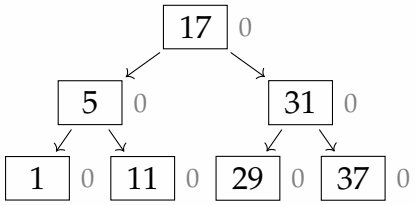

*Nach dem Einfügen von „31“*:



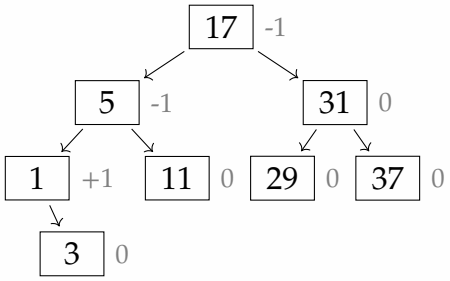

*Nach der Linksrotation*:




*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.

```java
public class AVLBaum {

}
```

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

# Thema Nr. 2

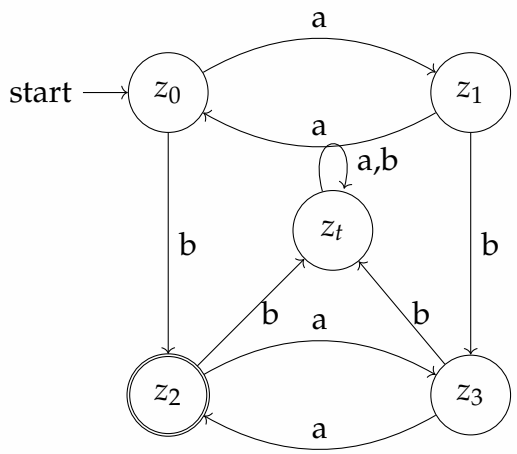
**Aufgabe 1 [Alphabet ab]**

# Aufgabe 1

(a) Gegeben ist die folgende Sprache L1 über dem Alphabet $\Sigma = \{a, b\}$:
$L1 = \{w \in \Sigma^* \mid \text{die Anzahl der } a \text{ in } w \text{ ist gerade und } b \text{ kommt in } w \text{ genau einmal vor}\}$.

(i) Geben Sie einen deterministischen endlichen Automaten an, der die Sprache $L1$ akzeptiert.

Lösungsvorschlag




Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Ahfqe3eb7

(ii) Geben Sie einen regulären Ausdruck an, der die Sprache $L1$ beschreibt.

Lösungsvorschlag

(aa)*(b|aba)(aa)*

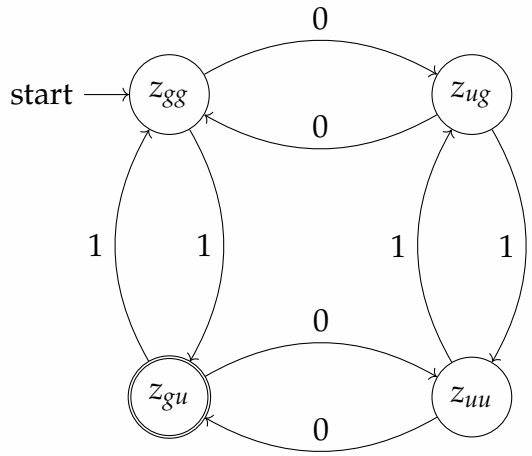
(b) Die folgende Sprache $L2$ ist eine Erweiterung von $L1$:
$L1 = \{w \in \Sigma^* \mid \text{die Anzahl der } a \text{ in } w \text{ ist gerade und die Anzahl der } b \text{ in } w \text{ ist ungerade}\}$.

(i) Geben Sie einen deterministischen endlichen Automaten an, der die Sprache $L2$ akzeptiert.

Lösungsvorschlag

*a*

gu = gerade Anzahl a's, ungerade Anzahl b's
ug = ungerade Anzahl a's, gerade Anzahl b's




Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Af0vcjys9

[a] `https://www.informatik.uni-hamburg.de/TGI/lehre/vl/SS14/FGI1/Folien/fgi1_v2_handout.pdf`

(ii) Geben Sie eine rechtslineare Grammatik an, die die Sprache $L2$ erzeugt.

Lösungsvorschlag

$$\mathrm{P} = \Big\{ \quad A \to aB \mid bD \mid b \quad B \to bC \mid aA \quad C \to aD \mid a \mid bB \quad D \to bA \mid aC \quad \Big\}$$

https://flaci.com/Ahfqe3eb7