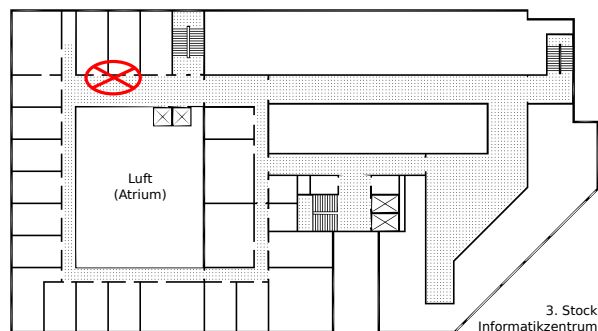


Hausaufgabenblatt 4

Abgabe der Lösungen bis zum Dienstag, den 07.01.2025 um 13:00 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet.

Schreibe auf die Abgabe unbedingt deinen Namen, Matrikel- und Gruppennummer! Mehrere Blätter tackern!



Hausaufgabe 1 (Binäre Suchbäume):

(3 Punkte)

Betrachte einen binären Suchbaum T , in welchem kein Element doppelt vorhanden ist. Angenommen, die Suche nach einem Element mit Wert k in T endet in einem Blatt. Dieser Suchpfad von der Wurzel zum Blatt teilt die Elemente von T in die drei Mengen A_L , A_P und A_R , die links vom, auf dem, bzw. rechts vom Suchpfad liegen.

Zeige oder widerlege: Für alle $a_1 \in A_L, a_2 \in A_P, a_3 \in A_R$ gilt $a_1 \leq a_2 \leq a_3$.

(Hinweis: Achtung, ist beispielsweise A_R leer (besitzt also keine Elemente), so gilt $a_2 \leq a_3$.)

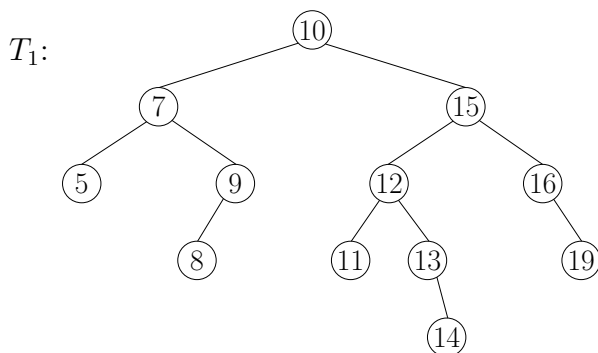
Hausaufgabe 2 (AVL-Bäume):

(3+2+2+4 Punkte)

Betrachte in den Aufgabenteilen a) und b) den Baum, der in der jeweiligen Abbildung dargestellt ist. Führe die Operation des jeweiligen Aufgabenteils und die damit verbundenen Restrukturierungsmaßnahmen zum Erhalt der AVL-Eigenschaft auf dem entsprechenden Baum aus. Zeichne dabei das Resultat nach jeder einzelnen ausgeführten Operation INSERT, DELETE und RESTRUCTURE in einen separaten Baum.

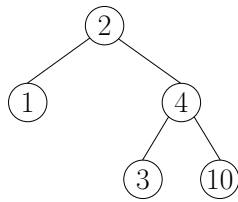
(Hinweis: AVL-Bäume werden in den Vorlesungen 15 und 16 sowie in Übung 4 behandelt.)

a) DELETE($T_1, 5$)



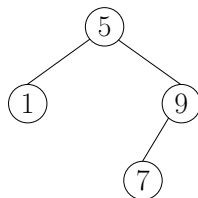
b) $\text{INSERT}(T_2, 11)$

T_2 :



c) Für welche Werte von $x \in \{2, 3, 4, 6, 8, 10\}$ führt die Operation $\text{INSERT}(T_3, x)$ zu einer Restrukturierung des Baumes? (Ohne Begründung.)

T_3 :



d) Zeige mit vollständiger Induktion, dass ein AVL-Baum der Höhe h mindestens $F_{h+2} - 1$ Knoten enthält.

(Hinweis: F_n beschreibt die n -te Fibonacci-Zahl mit $F_0 = 0$, $F_1 = 1$ und $F_n = F_{n-1} + F_{n-2}$.)

Hausaufgabe 3 (Einfach-verkettete Listen):

(6 Punkte)

Aus der Vorlesung kennen wir bereits die Operationen `LIST-INSERT` und `LIST-DELETE` für doppelt verkettete Listen. Wir wollen diese Operationen nun für einfach verkettete Listen betrachten, bei denen jedes Element lediglich seinen Nachfolger kennt:

- Ist die Funktion `LIST-INSERT`, wie sie aus der Vorlesung bekannt ist, mit Laufzeit $\mathcal{O}(1)$ realisierbar? Begründe!
- Ist die Funktion `LIST-DELETE`, wie sie aus der Vorlesung bekannt ist, mit Laufzeit $\mathcal{O}(1)$ realisierbar? Begründe!

(Hinweis: Falls die Funktion in der gegebenen Laufzeit realisierbar ist, gib eine entsprechende Funktion in Pseudocode an.)