

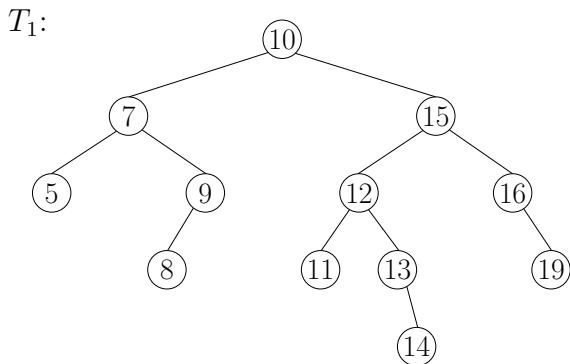
Hausaufgabenblatt 4

Abgabe der Lösungen muss bis zum 24.01.2022 um 11:00 Uhr erfolgen. Lösungen müssen per Mail mit einer pdf-Datei (Name der Datei „blatt_[nr]_[name]_[matrikel].pdf“) an den jeweiligen Tutor geschickt werden. Email-Adressen sind unter <https://www.ibr.cs.tu-bs.de/alg/index.html> zu finden.

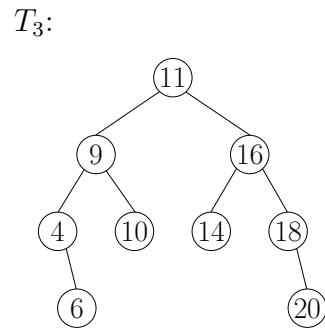
Beachte: Bei der Bearbeitung der Hausaufgaben gelten folgenden Richtlinien:
<https://www.ibr.cs.tu-bs.de/alg/Merkzettel/homework-booklet.pdf>

Hausaufgabe 1 (AVL-Bäume): (3+2+2+2+4 Punkte)
 Betrachte in den Aufgabenteilen a) bis d) 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:

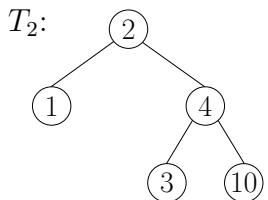
a) $\text{DELETE}(T_1, 5)$



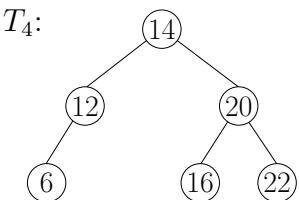
c) $\text{INSERT}(T_3, 5)$



b) $\text{DELETE}(T_2, 1)$



d) $\text{INSERT}(T_4, 4)$



e) 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 2 (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 3 (Einfach-Verkettete Listen): (2+2 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 mit Laufzeit $\mathcal{O}(1)$ realisierbar? Begründe!
- Ist die Funktion LIST-DELETE mit Laufzeit $\mathcal{O}(1)$ realisierbar? Begründe!

(Hinweis: Es ist kein Pseudocode erforderlich.)