

Hausaufgabenblatt 4

Abgabe der Lösungen muss bis zum 18.01.2021 um 14:00 Uhr erfolgen. Lösungen müssen per Mail mit einer pdf-Datei (Name der Datei „blatt_[nr]_[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/courses/ws2021/aud/HA-Hinweise.pdf>

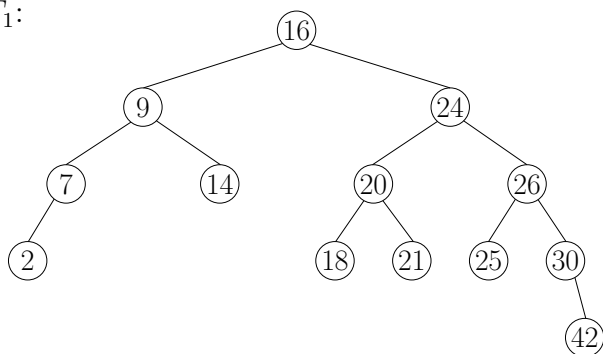
Hausaufgabe 1 (AVL-Bäume):

(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:

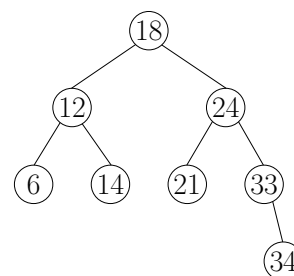
a) DELETE($T_1, 14$)

T_1 :



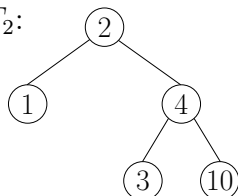
c) INSERT($T_3, 40$)

T_3 :



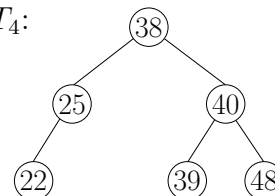
b) DELETE($T_2, 1$)

T_2 :



d) INSERT($T_4, 23$)

T_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 (Algorithmenentwurf):

(3+4 Punkte)

In dieser Aufgabe wollen wir einen Algorithmus entwickeln, der in linearer Zeit überprüft, ob ein gegebener binärer Suchbaum B ein AVL-Baum ist.

Bitte beachte bei der Bearbeitung die Hinweise zu Pseudocode unter <https://www.ibr.cs.tu-bs.de/courses/ws2021/aud/uebungen/PseudocodeZettel.pdf>. Einfache mathematische Funktionen wie die Maximumsfunktion $\max(a, b)$ oder der Betrag $\text{abs}(x)$ (oder $|x|$) dürfen ohne Definition direkt verwendet werden.

- a) Entwirf zunächst einen rekursiven Algorithmus mit dem Namen `Label(v)`, der jedem Knoten w in B mit $\text{Wurzel}(B) := v$ ein Label $h(w)$ zuweist, welches die Höhe von v in B beschreibt. Die Laufzeit darf $O(n)$ nicht überschreiten. Der Algorithmus darf außerdem nicht länger als 12 Zeilen sein (end if, end while, end function, etc. ausgenommen).
(Hinweis: Es darf angenommen werden, dass `LABEL(Wurzel(B))` aufgerufen wird.)
- b) Entwirf einen rekursiven Algorithmus mit dem Namen `isAVL(v)`, der überprüft, ob B mit $\text{Wurzel}(B) := v$ ein AVL-Baum ist. Die Laufzeit darf $O(n)$ nicht überschreiten. Der Algorithmus darf außerdem nicht länger als 15 Zeilen sein.
(Hinweis: Es kann angenommen werden, dass bereits jedem Knoten ein Label zugewiesen wurde, welches die Höhe des Knotens im AVL-Baum beschreibt. Das Label für einen Knoten v lässt sich dann mit $h(v)$ abrufen.)