

Übungsblatt 4

Datenstrukturen und Algorithmen (SS 2022)

Abgabe: Montag, 16.05.2022, 15:30 Uhr — Besprechung: ab Montag, 23.05.2022

Abgabevorschriften: Die Aufgaben auf diesem Blatt sind unter Einhaltung der Abgabevorschriften¹ zu lösen und abzugeben.

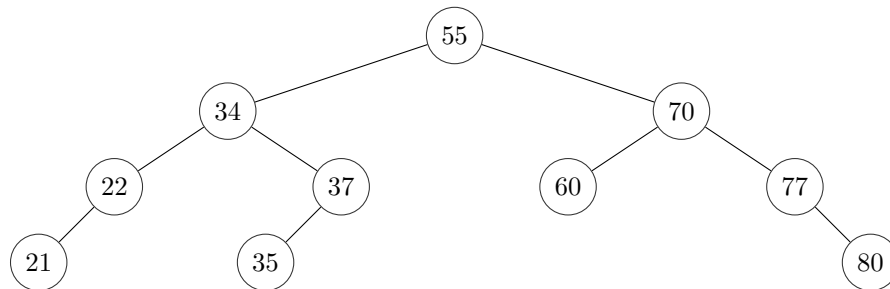
Lernziele: Nach dem Tutorium zu diesem Blatt sollten Sie folgende Lernziele erreicht haben. Wenn nicht, zögern Sie nicht, ihre:n Tutor:in anzusprechen um die Lücken zu füllen, Unklarheiten zu klären oder Fragen zu beantworten.

- Sie können AVL-Bäume erstellen, updaten, ausbalancieren, und in Java implementieren.
- Sie können 2-3-4-Bäume erstellen, updaten, ausbalancieren, in Rot-Schwarz-Bäume umwandeln, und in Java implementieren.
- Sie können Rot-Schwarz-Bäume erstellen, updaten, ausbalancieren, in 2-3-4-Bäume umwandeln, und in Java implementieren.
- Sie können B-Bäume erstellen, updaten, ausbalancieren, und in Java implementieren.
- Sie kennen die Eigenheiten, Unterschiede, Vorteile und Nachteile der vorgestellten ausbalancierten Bäume und können bei einem gegebenen Szenario eine geeignete Datenstruktur daraus auswählen.

Punkte: Dieses Übungsblatt beinhaltet 5 Aufgaben mit einer Gesamtzahl von 30 Punkten. Zum Bestehen werden also 15 Punkte benötigt.

Aufgabe 1 AVL-Bäume [Punkte: 6]

Gegeben sei der AVL-Baum AVL_1 :



- (a) (1 Punkte) Geben Sie für jeden Knoten aus dem ursprünglichen Baum AVL_1 den Wert der AVL-Balance an.
- (b) (2 Punkte) Erstellen Sie folgende AVL-Bäume durch Einfügen von Werten in den Baum AVL_1 :
- AVL_2 durch Einfügen von 64 in AVL_1
 - AVL_3 durch Einfügen von 20 in AVL_1
 - AVL_4 durch Einfügen von 36 in AVL_1
 - AVL_5 durch Einfügen von 78 in AVL_1

Verwenden Sie Zwischenergebnisse, um zu zeigen welche etwaigen Schritte notwendig sind, um die AVL-Balance wiederherzustellen. Geben Sie auch explizit an, welche Rotation um welchen Knoten durchgeführt wurde.

¹https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_file_2904210.html

(c) (2 Punkte) Erstellen Sie folgende *AVL-Bäume*:

- Entfernen Sie nacheinander 21 und 22 aus AVL_1 , um AVL_6 zu konstruieren
- Entfernen Sie nacheinander 80, 77 und 60 aus AVL_1 , um AVL_7 zu konstruieren

Verwenden Sie Zwischenergebnisse, um zu zeigen welche etwaigen Schritte notwendig sind, um die AVL-Balance wiederherzustellen. Geben Sie auch explizit an, welche Rotation um welchen Knoten durchgeführt wurde.

(d) (1 Punkte) 🦋 Was ist die maximale Höhe eines *AVL-Baums* mit 7 Knoten? Begründen Sie Ihre Antwort. Eine Antwort ohne Begründung wird mit null Punkten bewertet.

Aufgabe 2 Impl AVL-Bäume implementieren [Punkte: 8]

In Aufgabe 1 haben Sie sich bereits auf theoretischer Ebene mit AVL-Bäumen auseinandergesetzt. In dieser Aufgabe sollen Sie diese Konzepte in die Praxis umsetzen, indem Sie einen AVL-Baum in Java implementieren. Hierfür sind im Eclipse-Projekt folgende Codefragmente gegeben:

- Schnittstelle `IAVLNode` und unvollständige Klasse `AVLNode`
- Schnittstelle `IAVLTree` und unvollständige Klasse `AVLTree`

(a) (3 Punkte) Implementieren Sie zunächst die Knotenklasse `AVLNode`, die die Schnittstelle `IAVLNode` implementiert.

Hinweis: Die Knotenklasse muss einen Standard-Konstruktor (d.h. Konstruktor ohne Parameter) enthalten.

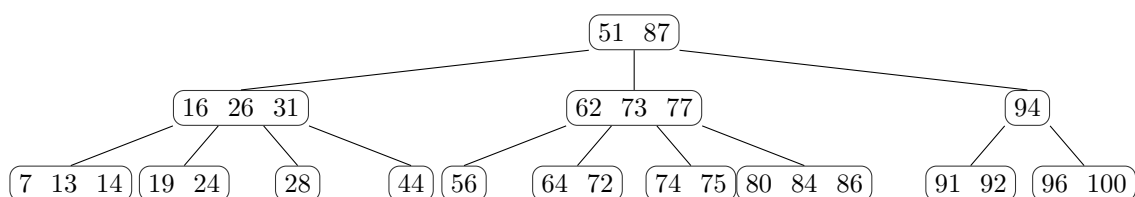
(b) (5 Punkte) Implementieren Sie nun die Klasse `AVLTree`, die die Schnittstelle `IAVLTree` implementiert.

Hinweis: Die Klasse muss einen Standard-Konstruktor (d.h. Konstruktor ohne Parameter) enthalten.

Tipp: Tests helfen dabei, die Funktionalität von Code zu überprüfen und so Fehler zu erkennen. Hierfür können Sie JUnit Tests in den Programmierprojekten erstellen. Um die Korrektheit Ihrer Implementierung zu überprüfen, können Sie diesmal auch Aufgabe 1a mit Hilfe Ihres Programms lösen und mit Ihrer Lösung vergleichen.

Aufgabe 3 2-3-4-Bäume [Punkte: 5]

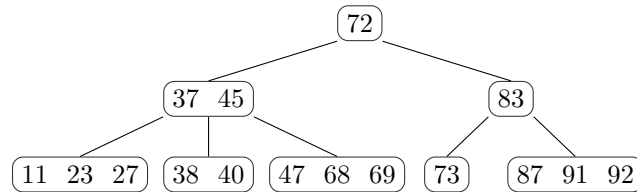
(a) (3 Punkte) Gegeben sei der 2-3-4-Baum ZDV_1 :



Fügen Sie in ZDV_1 nacheinander die Werte 15, 74, 29, 78 ein. Geben Sie alle Zwischenschritte an und erläutern Sie diese kurz. Wandeln Sie 4-Knoten nicht vorsorglich um.

(b) (1 Punkte) 🦋 Wie viele Knoten können höchstens in einem 2-3-4-Baum der Höhe 3 enthalten sein? Begründen Sie Ihre Antwort. Eine Antwort ohne Begründung wird mit null Punkten bewertet.

(c) (1 Punkte) Gegeben sei der 2-3-4-Baum ZDV_2 :

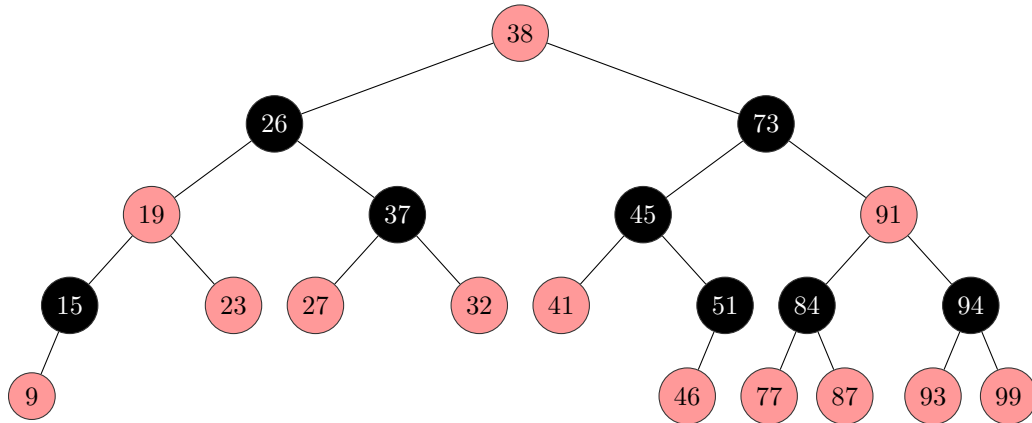


Welche der folgenden Sequenzen kann **keine** Einfügereihenfolge der Schlüssel in ZDV_2 sein, mit der ZDV_2 entstanden sein könnte? Begründen Sie Ihre Antwort. Eine Antwort ohne Begründung wird mit null Punkten bewertet.

- 72, 37, 83, 11, 40, 45, 68, 23, 27, 38, 47, 69, 73, 87, 91, 92
- 37, 92, 72, 69, 73, 11, 40, 83, 45, 23, 87, 68, 27, 47, 91, 38
- 11, 91, 72, 68, 73, 83, 45, 40, 87, 92, 37, 23, 27, 38, 47, 69
- 72, 83, 37, 73, 87, 91, 92, 11, 23, 27, 38, 40, 45, 47, 68, 69

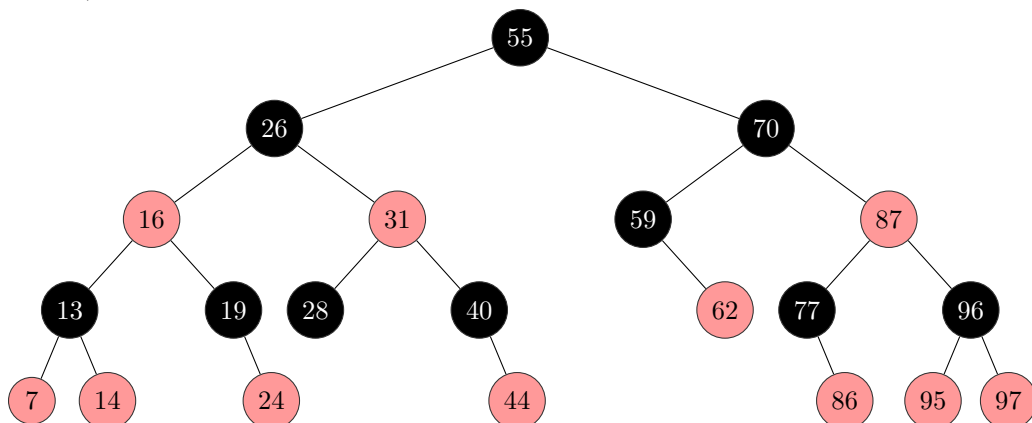
Aufgabe 4 Rot-Schwarz-Bäume [Punkte: 4]

(a) (2 Punkte) Gegeben ist der folgende Baum RS_1 :



Auch wenn seine Knoten gefärbt sind, ist er aus mehreren Gründen kein Rot-Schwarz-Baum. Markieren Sie die entsprechenden Stellen und begründen Sie, warum RS_1 deswegen kein Rot-Schwarz-Baum ist.

(b) (1 Punkte) Gegeben ist der folgende Rot-Schwarz-Baum RS_2 :

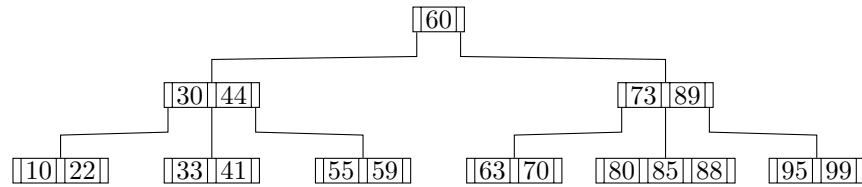


Wandeln Sie den ursprünglichen Rot-Schwarz-Baum RS_2 in einen 2-3-4-Baum um.

(c) (1 Punkte) Fügen Sie in RS_2 den Wert 46 ein. Gehen Sie hierbei nach der Top-Down-Variante vor. Geben Sie Zwischenschritte an und erläutern Sie diese kurz.

Aufgabe 5 B-Bäume [Punkte: 7]

- (a) (3 Punkte) Fügen Sie in einem leeren B-Baum mit $m = 2$ der Reihe nach die Zahlen 12, 16, 14, 9, 20, 15, 2, 22, 30, 1, 23, 11, 52, 7, 4, 27 und 5 ein. Geben Sie Zwischenschritte, in denen ein Split notwendig ist, an und erläutern Sie diese kurz.
- (b) (2 Punkte) Gegeben sei der folgende B-Baum B_1 . Entfernen Sie nacheinander die Elemente 30 und 99 und zeichnen Sie B_1 nach jedem Schritt.



- (c) (1 Punkte) 🦋 Wieviele Einträge muss ein B-Baum mit der Ordnung $m = 3$ und drei Ebenen mindestens enthalten? Begründen Sie Ihre Antwort. Eine Antwort ohne Begründung wird mit null Punkten bewertet.
- (d) (1 Punkte) 🦋 B-Bäume werden in der Praxis vor allem für sehr große Datenmengen mit vielen Elementen pro Seite eingesetzt. Nehmen Sie an, dass sich in dem B-Baum B_2 75 - 150 Elemente auf einer Seite befinden, die Ordnung von B_2 also 75 beträgt ($m = 75$). Wie viele Seitenzugriffe sind bei einer Menge von 1.000.000.000 Elementen im schlechtesten Fall nötig, um ein Element zu finden? Geben Sie Ihre Rechnung an und begründen Sie diese. Eine Antwort ohne Begründung wird mit null Punkten bewertet.