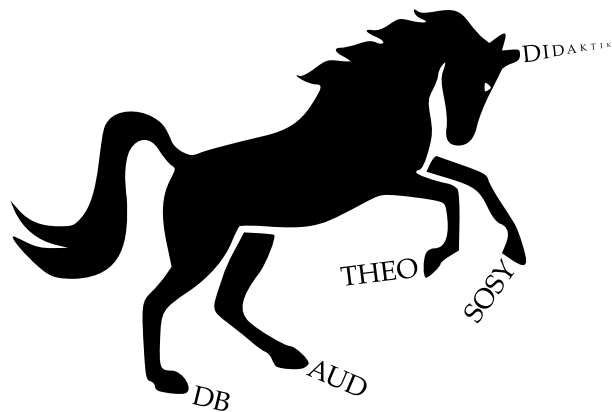


Die komplette Sammlung

Alle Aufgaben



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Inhaltsverzeichnis

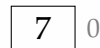
| | |
|---|----|
| 66115 / 2014 / Frühjahr / Thema 1 / Aufgabe 3 | 3 |
| 66115 / 2017 / Frühjahr / Thema 2 / Aufgabe 8 | 7 |
| 66115 / 2019 / Frühjahr / Thema 2 / Aufgabe 7 | 11 |
| 66116 / 2016 / Frühjahr / Thema 2 / Teilaufgabe 2 / Aufgabe 1 | 14 |
| 66116 / 2017 / Frühjahr / Thema 1 / Teilaufgabe 1 / Aufgabe 2 | 16 |

66115 / 2014 / Frühjahr / Thema 1 / Aufgabe 3

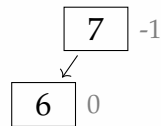
- (a) Fügen Sie die Zahlen (7, 6, 2, 1, 5, 3, 8, 4) in dieser Reihenfolge in einen anfangs leeren AVL Baum ein. Stellen Sie die AVL Eigenschaft ggf. nach jedem Einfügen mit geeigneten Rotationen wieder her. Zeichnen Sie den AVL Baum einmal vor und einmal nach jeder einzelnen Rotation.

Lösungsvorschlag

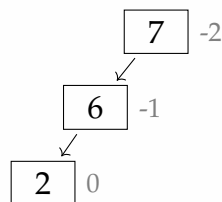
Nach dem Einfügen von „7“:



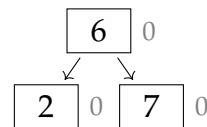
Nach dem Einfügen von „6“:



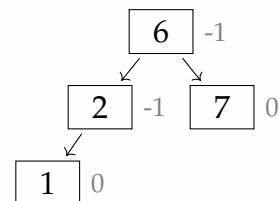
Nach dem Einfügen von „2“:



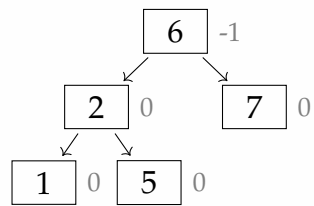
Nach der Rechtsrotation:



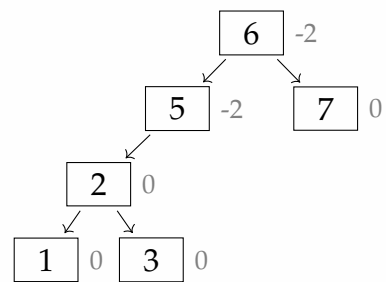
Nach dem Einfügen von „1“:



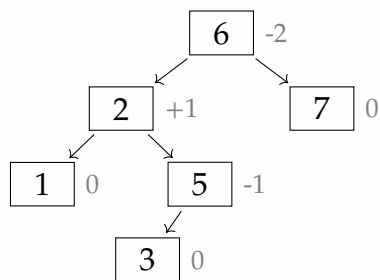
Nach dem Einfügen von „5“:



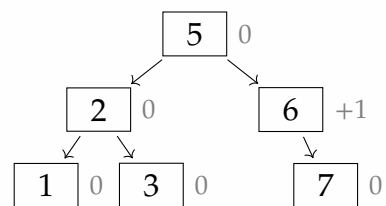
Nach der Linksrotation:



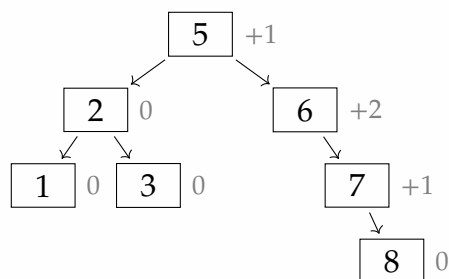
Nach dem Einfügen von „3“:



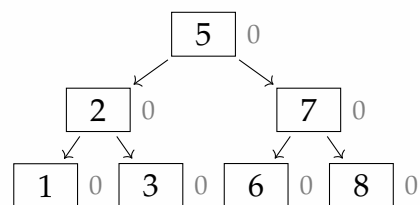
Nach der Rechtsrotation:



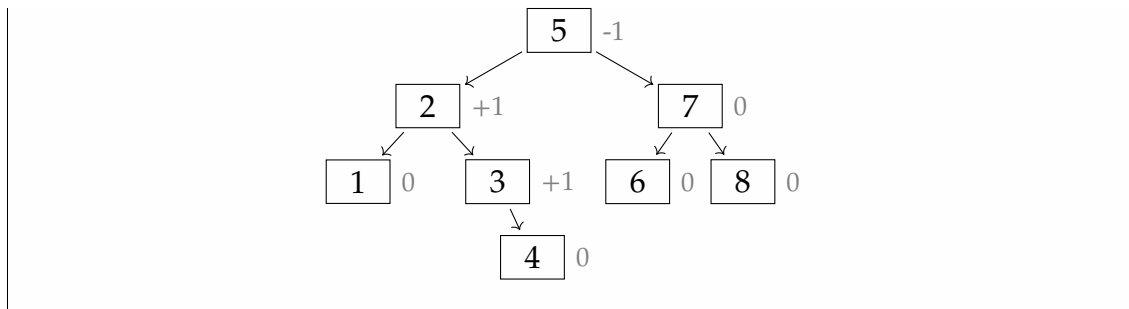
Nach dem Einfügen von „8“:



Nach der Linksrotation:

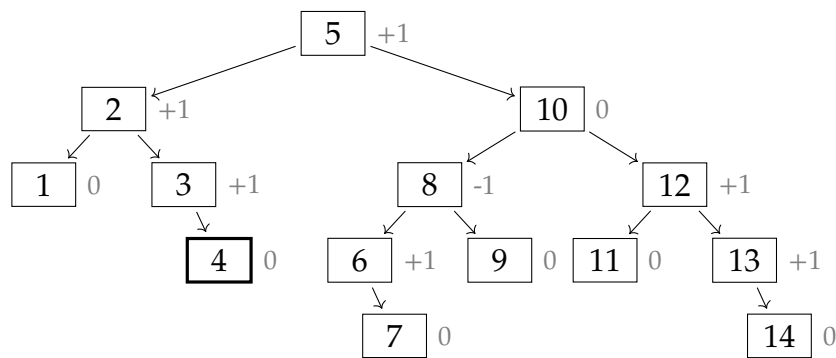


Nach dem Einfügen von „4“:



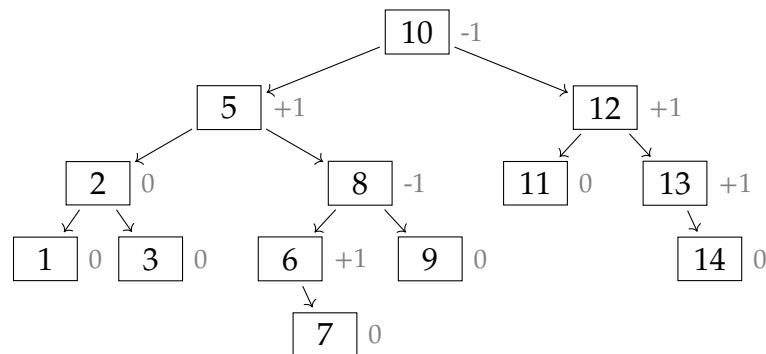
- (b) Entfernen Sie den jeweils markierten Knoten aus den folgenden AVL-Bäumen. Stellen Sie die AVL-Eigenschaft ggf. durch geeignete Rotationen wieder her. Zeichnen Sie nur den resultierenden Baum.

(i) Baum 1:

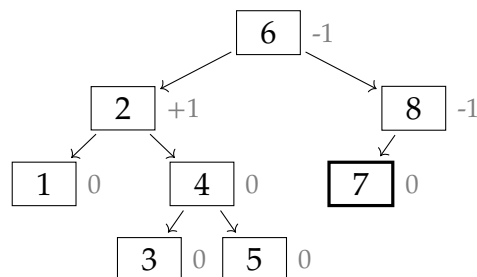


Lösungsvorschlag

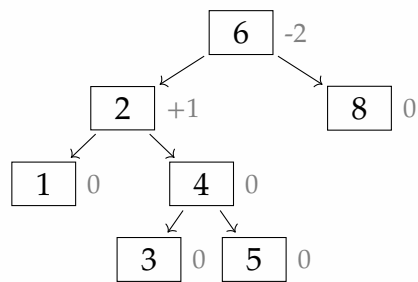
Nach dem Löschen von „4“:



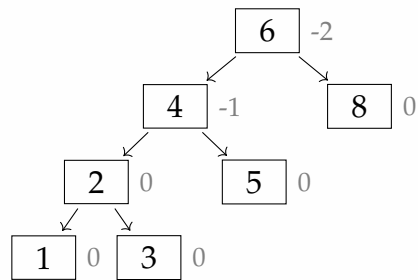
(ii) Baum 2:



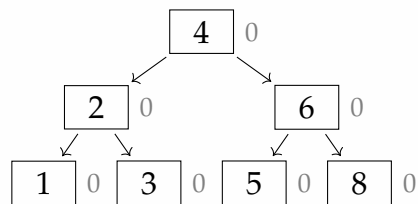
Nach dem Löschen von „7“:



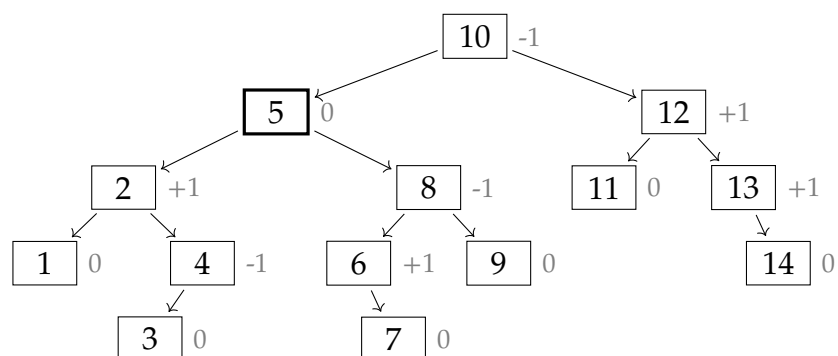
Nach der Linksrotation:



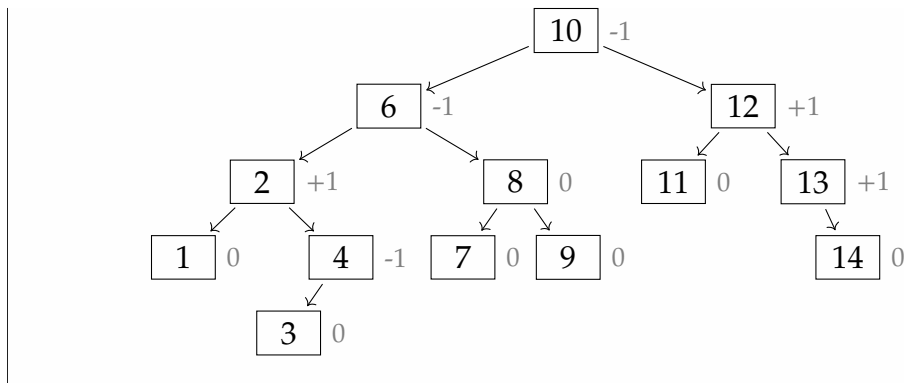
Nach der Rechtsrotation:



(iii) Baum 3:



Nach dem Löschen von „5“:

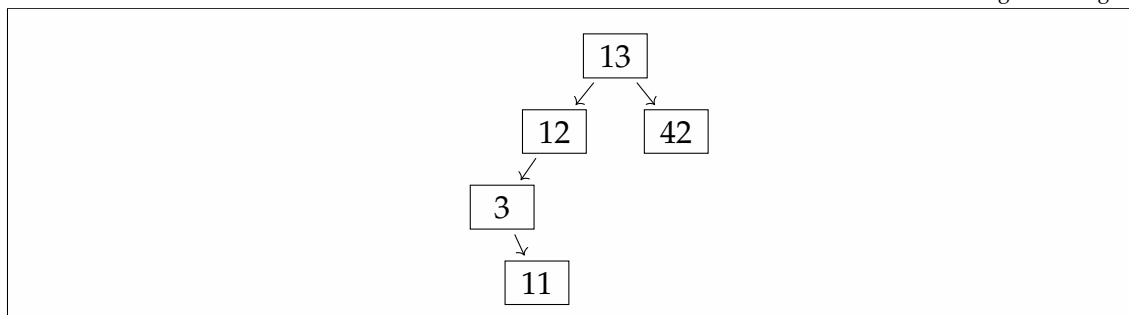


AVL-Baum
Halbe (Heap)
Binärbaum

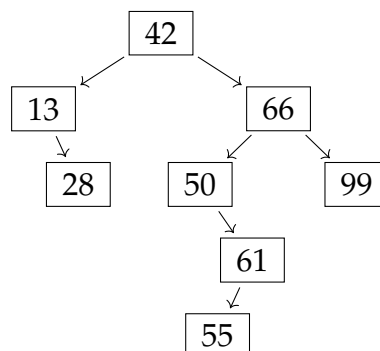
66115 / 2017 / Frühjahr / Thema 2 / Aufgabe 8

- (a) Fügen Sie die Zahlen 13, 12, 42, 3, 11 in der gegebenen Reihenfolge in einen zunächst leeren binären Suchbaum mit aufsteigender Sortierung ein. Stellen Sie nur das Endergebnis dar.

Lösungsvorschlag

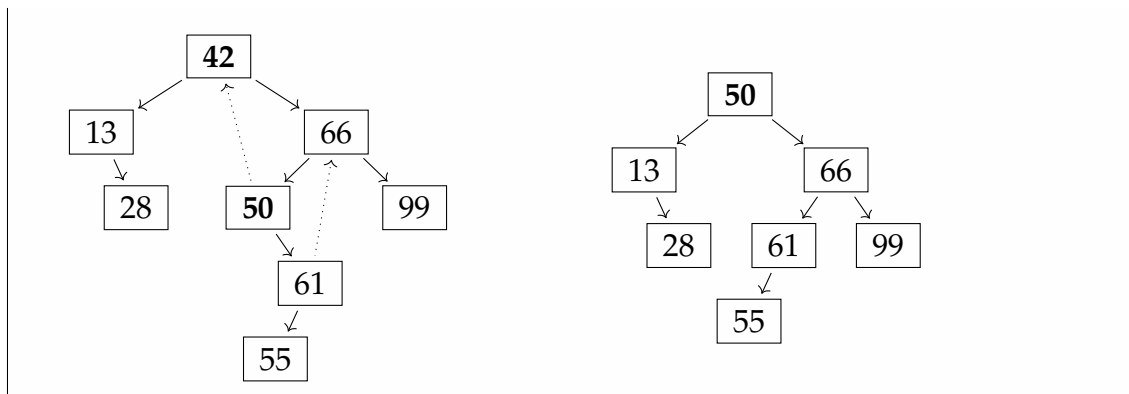


- (b) Löschen Sie den Wurzelknoten mit Wert 42 aus dem folgenden *binären* Suchbaum mit aufsteigender Sortierung und ersetzen Sie ihn dabei durch einen geeigneten Wert aus dem *rechten* Teilbaum. Lassen Sie möglichst viele Teilbäume unverändert und erhalten Sie die Suchbaumeigenschaft.

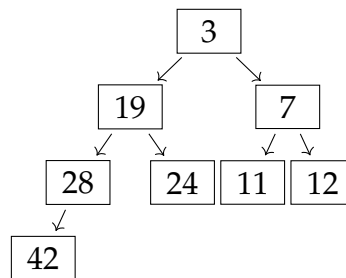


Lösungsvorschlag



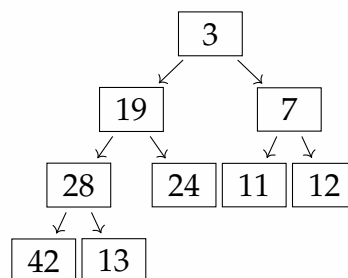


- (c) Fügen Sie einen neuen Knoten mit dem Wert 13 in die folgende Min-Halbe ein und stellen Sie anschließend die Halden-Eigenschaft vom neuen Blatt aus beginnend wieder her, wobei möglichst viele Knoten der Halde unverändert bleiben und die Halde zu jedem Zeitpunkt links-vollständig sein soll. Geben Sie nur das Endergebnis an.

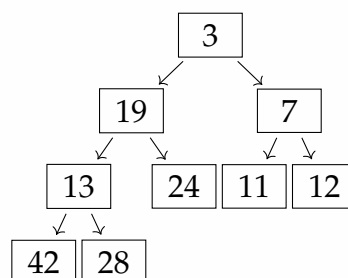


Lösungsvorschlag

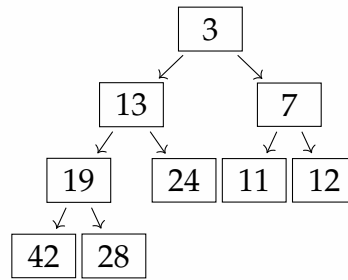
Nach dem Einfügen von „13“:



Nach dem Vertauschen von „13“ und „28“:



Nach dem Vertauschen von „13“ und „19“:

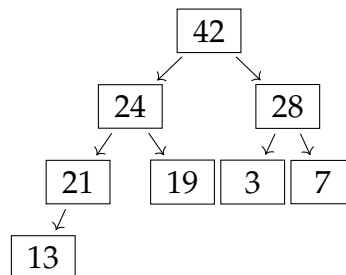


- (d) Geben Sie für die ursprüngliche Min-Halde aus Teilaufgabe c) (lösche den neu eingefügten Knoten mit dem Wert 13) die Feld-Einbettung (Array-Darstellung) an.

Lösungsvorschlag

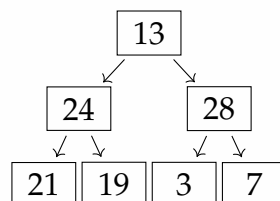
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|----|---|----|----|----|----|----|
| 3 | 19 | 7 | 28 | 24 | 11 | 12 | 42 |

- (e) Löschen Sie den Wurzelknoten mit Wert 42 aus der folgenden Max-Halde und stellen Sie anschließend die Halden-Eigenschaft ausgehend von einer neuen Wurzel wieder her, wobei möglichst viele Knoten der Halde unverändert bleiben und die Halde zu jedem Zeitpunkt links-vollständig sein soll. Geben Sie nur das Endergebnis an.

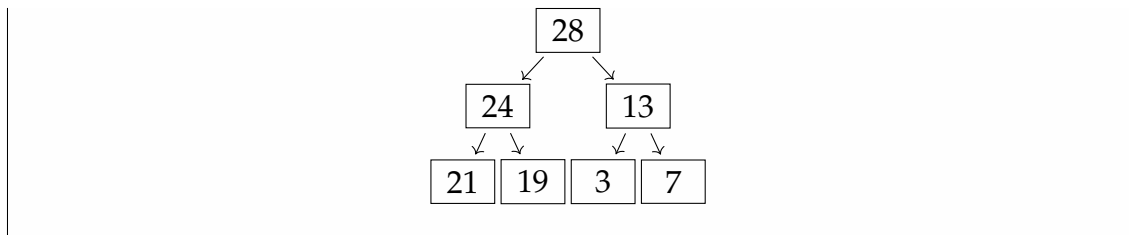


Lösungsvorschlag

Nach dem Ersetzen von „42“ mit „13“:

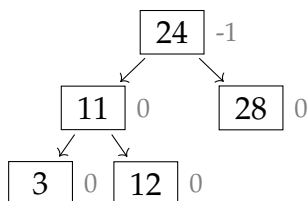


Nach dem Vertauschen von „13“ und „28“:



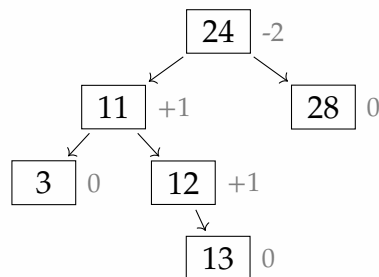
- (f) Fügen Sie in jeden der folgenden AVL-Bäume mit aufsteigender Sortierung jeweils einen neuen Knoten mit dem Wert 13 ein und führen Sie anschließend bei Bedarf die erforderliche(n) Rotation(en) durch. Zeichnen Sie den Baum vor und nach den Rotationen.

(i) AVL-Baum A

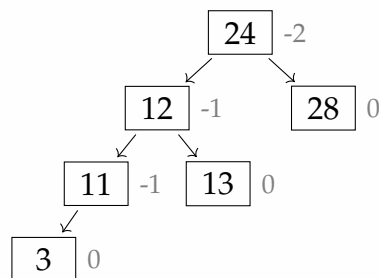


Lösungsvorschlag

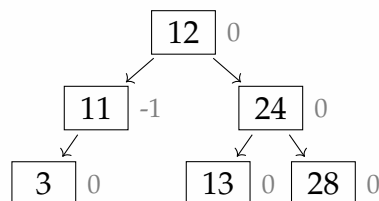
Nach dem Einfügen von „13“:



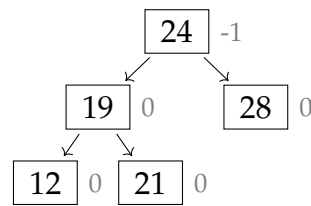
Nach der Linksrotation:



Nach der Rechtsrotation:

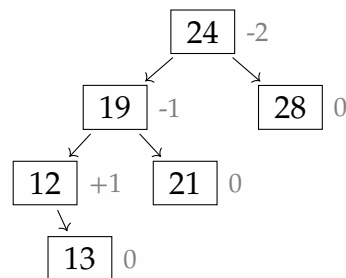


(ii) AVL-Baum B

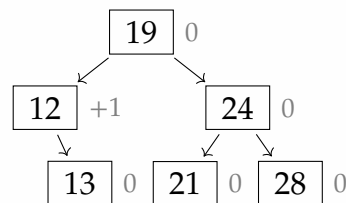


Lösungsvorschlag

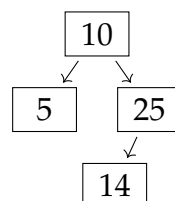
Nach dem Einfügen von „13“:



Nach der Rechtsrotation:

**66115 / 2019 / Frühjahr / Thema 2 / Aufgabe 7**

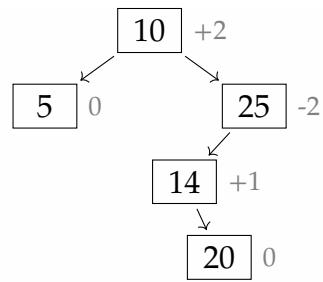
Fügen Sie (manuell) nacheinander die Zahlen 20, 31, 2, 17, 7 in folgenden AVL-Baum ein. Löschen Sie anschließend aus dem entstandenen Baum nacheinander 14 und 25.



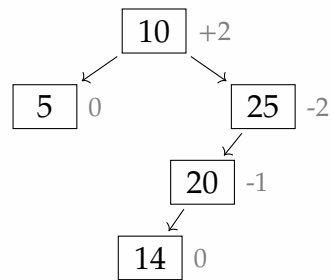
Zeichnen Sie jeweils direkt nach jeder einzelnen Operation zum Einfügen oder Löschen eines Knotens, sowie nach jeder elementaren Rotation den entstehenden Baum. Insbesondere sind evtl. anfallende Doppelrotationen in zwei Schritten darzustellen. Geben Sie zudem an jedem Knoten die Balancewerte an.

Lösungsvorschlag

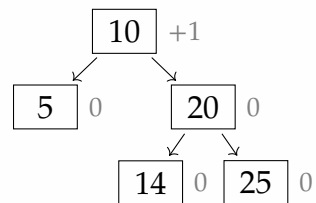
Nach dem Einfügen von „20“:



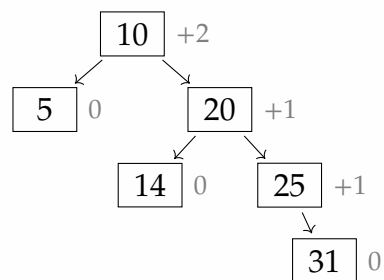
Nach der Linksrotation:



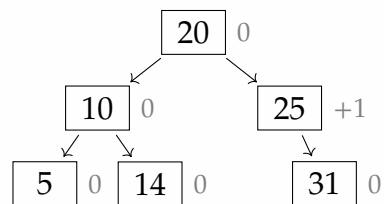
Nach der Rechtsrotation:



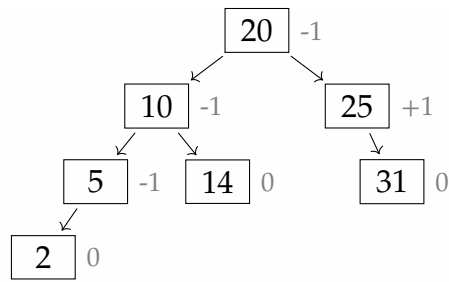
Nach dem Einfügen von „31“:



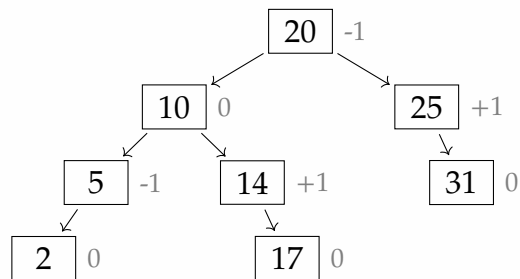
Nach der Linksrotation:



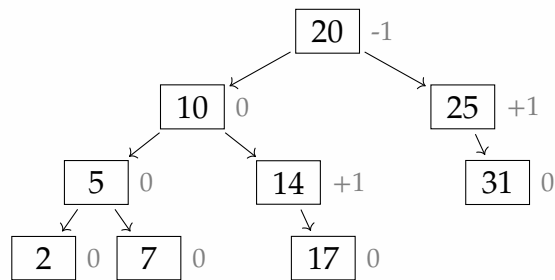
Nach dem Einfügen von „2“:



Nach dem Einfügen von „17“:

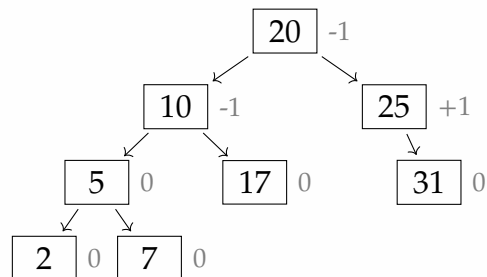


Nach dem Einfügen von „7“:

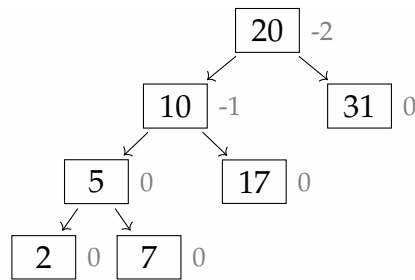


Löschen

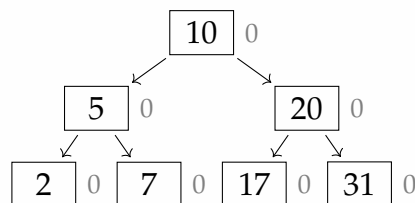
Nach dem Löschen von „14“:



Nach dem Löschen von „25“:



Nach der Rechtsrotation:



66116 / 2016 / Frühjahr / Thema 2 / Teilaufgabe 2 / Aufgabe 1

Ordnen Sie die folgenden Aussagen entsprechend ihres Wahrheitsgehaltes in einer Tabelle der folgenden Form an:

| Kategorie | WAHR | FALSCH |
|-----------|--------|--------|
| X | X1, X3 | X2 |
| Y | Y2 | Y1 |
| ... | ... | ... |

A Allgemein

- A1** Im Software Engineering geht es vor allem darum qualitativ hochwertige Software zu entwickeln.
- A2** Software Engineering ist gleichbedeutend mit Programmieren.

B Vorgehensmodelle

- B1** Die Erhebung und Analyse von Anforderungen sind nicht Teil des Software Engineerings.
- B2** Agile Methoden eignen sich besonders gut für die Entwicklung komplexer und sicherer Systeme in verteilten Entwicklerteams.
- B3** Das Spiralmodell ist ein Vorläufer sogenannter Agiler Methoden.

C Anforderungserhebung

- C1** Bei der Anforderungserhebung dürfen in keinem Fall mehrere Erhebungstechniken (z. B. Workshops, Modellierung) angewendet werden, weil sonst Widersprüche in Anforderungen zu Vorschein kommen könnten.

- C2** Ein Szenario beinhaltet eine Menge von Anwendungsfällen.
- C3** Nicht-funktionale Anforderungen sollten, wenn möglich, immer quantitativ spezifiziert werden.

Nicht-funktionale
Anforderungen
Entwurfsmuster
Schichtenarchitektur
Blackboard-Muster
Einbringen von
Abhängigkeiten
(Dependency Injection)
Sequenzdiagramm
Zustandsdiagramm Wissen
Komponentendiagramm
Modell-Präsentation-
Steuerung
(Model-View-Controller)
Einzelstück (Singleton)
Kommando (Command)
Validation
Verifikation

D Architekturmuster

- D1** Schichtenarchitekturen sind besonders für Anwendungen geeignet, in denen Performance eine wichtige Rolle spielt.
- D2** Das Black Board Muster ist besonders für Anwendungen geeignet, in denen Performance eine wichtige Rolle spielt.
- D3** „Dependency Injection“ bezeichnet das Konzept, welches Abhängigkeiten zur Laufzeit reglementiert.

E UML

- E1** Sequenzdiagramme beschreiben Teile des Verhaltens eines Systems.
- E2** Zustandsübergangsdiagramme beschreiben das Verhalten eines Systems.
- E3** Komponentendiagramme beschreiben die Struktur eines Systems.

F Entwurfsmuster

- F1** Das MVC Pattern verursacht eine starke Abhängigkeit zwischen Datenmodell und Benutzeroberfläche.
- F2** Das Singleton Pattern stellt sicher, dass es zur Laufzeit von einer bestimmten Klasse höchstens ein Objekt gibt.
- F3** Im Kommando Entwurfsmuster (engl. „Command Pattern“) werden Befehle in einem sog. Kommando-Objekt gekapselt, um sie bei Bedarf rückgängig zu machen.

G Testen

- G1** Validation dient der Überprüfung von Laufzeitfehlern.
- G2** Testen ermöglicht sicherzustellen, dass ein Programm absolut fehlerfrei ist.
- G3** Verifikation dient der Überprüfung, ob ein System einer Spezifikation entspricht.

Lösungsvorschlag

| Kategorie | WAHR | FALSCH |
|-----------|------------|-----------------------------------|
| A | A1 | A2 |
| B | B3 | B1, B2 |
| C | C3 | C1, C2 |
| D | D3 | D1, D2 |
| E | E1, E2, E3 | |
| F | F2, F3 | F1 |
| G | G3 | G1 ^a , G2 ^b |

^aValidierung: Prüfung der Eignung beziehungsweise der Wert einer Software bezogen auf ihren Einsatzzweck: „Wird das richtige Produkt entwickelt?“

^bEin Softwaretest prüft und bewertet Software auf Erfüllung der für ihren Einsatz definierten Anforderungen und misst ihre Qualität.

66116 / 2017 / Frühjahr / Thema 1 / Teilaufgabe 1 / Aufgabe 2

Konstruieren Sie einen B-Baum, dessen Knoten maximal 4 Einträge enthalten können, indem Sie der Reihe nach diese Suchschlüssel einfügen:

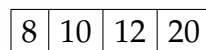
8, 10, 12, 20, 5, 30, 25, 11

Anschließend löschen Sie den Eintrag mit dem Suchschlüssel 8.

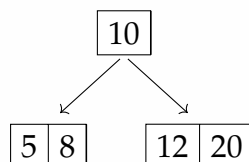
Zeigen Sie jeweils graphisch den entstehenden Baum nach relevanten Zwischenschritten; insbesondere nach Einfügen der 5 sowie nach dem Einfügen der 11 und nach dem Löschen der 8.

Lösungsvorschlag

- Schlüsselwert 8 (einfaches Einfügen)
- Schlüsselwert 10 (einfaches Einfügen)
- Schlüsselwert 12 (einfaches Einfügen)
- Schlüsselwert 20 (einfaches Einfügen)

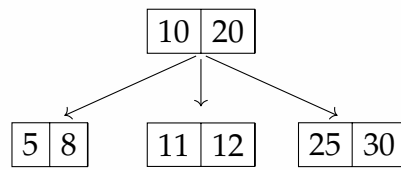


- Schlüsselwert 5 (Split)

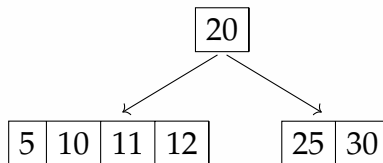


- Schlüsselwert 30 (einfaches Einfügen)
- Schlüsselwert 25 (einfaches Einfügen)

- Schlüsselwert 11 (Split)



- Löschen des Schlüsselwerts 8 (Mischen/Verschmelzen)



Index

Agile Methoden, 14

AVL-Baum, 3, 7, 11

B-Baum, 16

Binärbaum, 7

Blackboard-Muster, 15

Einbringen von Abhängigkeiten (Dependency Injection), 15

Einzelstück (Singleton), 15

Entwurfsmuster, 15

Halde (Heap), 7

Kommando (Command), 15

Komponentendiagramm, 15

Modell-Präsentation-Steuerung (Model-View-Controller), 15

Nicht-funktionale Anforderungen, 15

Schichtenarchitektur, 15

Sequenzdiagramm, 15

Software Engineering, 14

Spiralmodell, 14

Validation, 15

Verifikation, 15

Zustandsdiagramm Wissen, 15