Universität Stuttgart

Übungsblatt 3

Datenstrukturen und Algorithmen (SS 2022)

Abgabe: Montag, 09.05.2022, 15:30 Uhr — Besprechung: ab Montag, 16.05.2022

Abgabevorschriften: Die Aufgaben auf diesem Blatt sind unter Einhaltung der Abgabevorschriften[1] zu lösen und abzugeben.

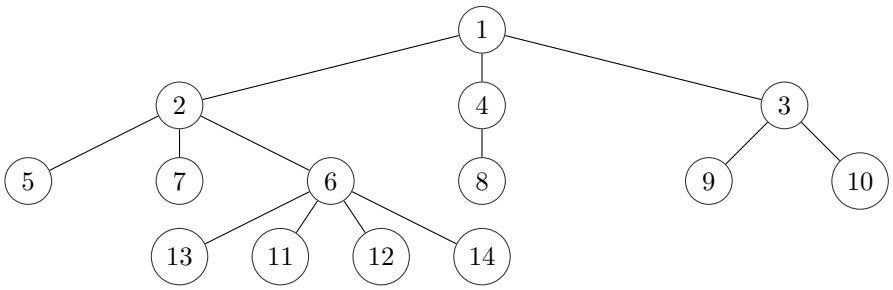
Lernziele: Nach dem Tutorium zu diesem Blatt sollten Sie folgende Lernziele erreicht haben. Wenn nicht, zögern Sie nicht, ihre:n Tutor:in anzusprechen um die Lücken zu füllen, Unklarheiten zu klären oder Fragen zu beantworten.

- Sie können einen gegebenen Baum mit Fachbegriffen beschreiben.
- Sie können Bäume, Binärbäume und Suchbäume als solche erkennen, sie erstellen, Knoten hinzufügen und Knoten löschen.
- Sie können Bäume in Level-, In-, Pre-, und Postorder traversieren und verstehen die Zusammenhänge zwischen diesen Traversierungen.
- Sie können mit Hilfe von Suchbäumen suchen.
- Sie können Binär- und Suchbäume sowie passende Iteratoren in Java implementieren und verwenden.

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

Aufgabe 1 Baumgrundschule *[Punkte: 3]*

Gegeben sei der Baum B_1:



(a) *(1 Punkt)* Beschreiben Sie B_1 mit den Fachbegriffen aus der Vorlesung. In Ihrer Beschreibung sollten die Schlüsselworte *Knoten*, *Kanten*, *Wurzel*, *Elternknoten*, *Kindknoten*, *innerer Knoten*, *Blatt*, *Pfad*, *Niveau*, *Höhe*, *n-är*, und *geordnet* vorkommen.

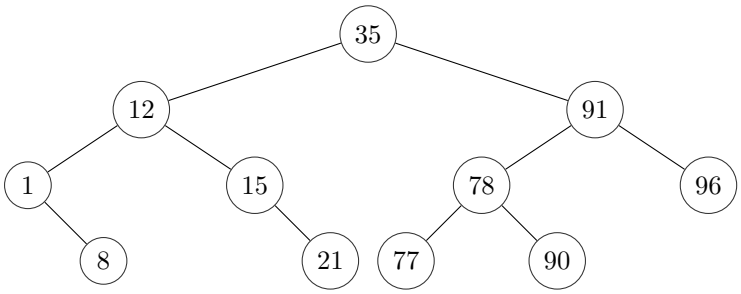
(b) *(1 Punkt)* Handelt es sich bei B_1 um einen Suchbaum? Begründen Sie Ihre Antwort.

(c) *(1 Punkt)* Wandeln Sie B_1 mit Hilfe des Binarisierungsalgorithmus der Vorlesung in einen binären Baum um.

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

Aufgabe 2 Binäre Suchbäume *[Punkte: 5]*

Gegeben sei der **binäre Suchbaum** B_2:



(a) *(1 Punkt)* Erstellen Sie den Baum B_3, indem Sie in B_2 die Werte 13 und 93 einfügen.

(b) *(1 Punkt)* Erstellen Sie den Baum B_4, indem Sie aus dem ursprünglichen Baum B_2 die Werte 15 und 91 entfernen. Ersetzen Sie, wo nötig, Knoten durch ihre **Inorder-Nachfolger**.

(c) *(1 Punkt)* Zeichnen Sie für diese Menge von Schlüsseln {10, 3, 21, 60, 6, 34, 40} einen binären Suchbaum mit Höhe 3 und einen binären Suchbaum mit Höhe 6. Falls dies nicht möglich ist, begründen Sie warum.

(d) *(1 Punkt)* Nehmen Sie an, Sie haben Zahlen zwischen 1 und 1000 in einem binären Suchbaum gespeichert und Sie haben in diesem Suchbaum nach der Zahl 363 gesucht. Dabei wurden mehrere Knoten überprüft. Welche der folgenden Sequenzen kann **keine** Knotenfolge sein, die während der Suche überprüft wurde? Begründen Sie Ihre Antwort. Eine Antwort ohne Begründung wird mit null Punkten bewertet.

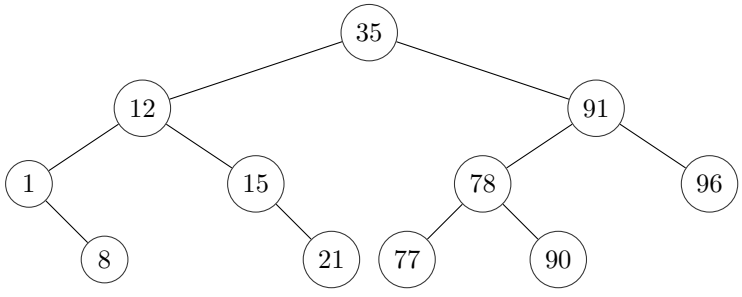
i. 2, 252, 401, 398, 330, 344, 397, 363
ii. 924, 220, 911, 244, 898, 258, 362, 363
iii. 925, 202, 911, 240, 912, 245, 363
iv. 2, 399, 387, 219, 266, 382, 381, 278, 363
v. 935, 278, 347, 621, 299, 392, 358, 363

(e) *(1 Punkt)* Telefonbücher bestehen insbesondere für größere Städte aus vielen Seiten und enthalten dementsprechend sehr viele Einträge. Die einzelnen Einträge in einem Telefonbuch können auch als Liste aufgefasst werden, welche alphabetisch geordnet sind. Würde es Sinn machen anstelle einer Liste einen Suchbaum zu verwenden? Begründen Sie. Eine Antwort ohne Begründung wird mit null Punkten bewertet.

Aufgabe 3 Baumtraversierung *[Punkte: 4]*

In dieser Aufgabe werden Sie mit systematischen Durchläufen aller Knoten eines Baumes arbeiten.

(a) *(2 Punkte)* Gegeben sei der binäre Suchbaum B_5:



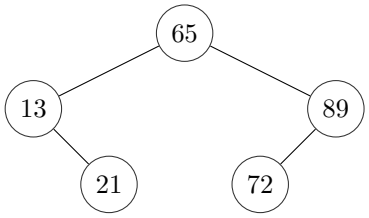
Geben Sie die Knoten des Baumes B_5 in Preorder- und Inorderreihenfolge an.

(b) *(1 Punkt)* Gegeben sind die Inordertraversierung 15, 47, 85, 67, 1, 86, 90, 8, 20, 69, 17, 12, 62 und die Postordertraversierung 15, 85, 47, 1, 86, 67, 8, 20, 17, 62, 12, 69, 90. Geben Sie den zugehörigen Binärbaum an.

Hinweis: Es handelt sich hier nicht notwendigerweise um einen Suchbaum!

(c) *(1 Punkt)* Ein Kommilitone hat behauptet, dass Pre- und Postorder-Reihenfolgen nicht zusammen ausreichen würden, um einen Binärbaum eindeutig zu bestimmen. Stimmt das? Wenn ja, erklären Sie, warum diese Behauptung stimmt. Wenn nein, geben Sie ein Gegenbeispiel an.
Hinweis: Es handelt sich hier nicht notwendigerweise um einen Suchbaum!

Aufgabe 4 Verständnisfragen Baumimplementierung *[Punkte: 4]*
Gegeben sei der binäre Suchbaum B_6:



(a) *(1 Punkt)* B_6 wurde erzeugt, indem ein Algorithmus über ein (nicht zwangsläufig sortiertes) Array iteriert und alle Elemente nacheinander in den Baum eingefügt hat ohne ihn auszubalancieren. Welche Arrays hätten als Eingabe für B_6 dienen können?

(b) *(1 Punkt)* Man kann Binärbäume auch in Arrays speichern, indem man das Array in Level-Order befüllt. Für B_6 würde das Array zum Beispiel so aussehen:

65	13	89	NULL	21	72	NULL

Wenn ein Binärbaumknoten am Array-Index b abgelegt wurde, an welchen Indizes können dann seine Kinder gefunden werden, wenn das Array mit **Index 0** startet? Begründen Sie Ihre Antwort.

(c) *(1 Punkt)* Entsprechend lässt sich für einen Koten an Index b auch die dazugehörige Baumebene berechnen. Welche Formel kann hier verwendet werden? Begründen Sie Ihre Antwort.
Hinweis: Aufrunden und Abrunden kann durch die Symbole $\lfloor\ \rfloor$ beziehungsweise $\lceil\ \rceil$ ausgedrückt werden, sofern nötig.

(d) *(1 Punkt)* Zeichnen Sie B_6 inklusive aller Pseudoknoten. Welche konkreten Vorteile bringen Pseudoknoten bei der Implementierung von Bäumen mit sich?

Aufgabe 5 Impl Suchbaum und Baumiteratoren *[Punkte: 14]*
In dieser Aufgabe sollen Sie eine Baumimplementierung ergänzen und Iteratoren für vier verschiedene Baumtraversierungen implementieren. Hierfür sind im Eclipse-Projekt folgende Codefragmente für binäre Suchbäume sowie Iteratoren gegeben:

- Schnittstelle `IBinaryTreeNode` und unvollständige Klasse `BinaryTreeNode`
- Schnittstelle `IBinarySearchTree` und unvollständige Klasse `BinarySearchTree`
- Schnittstelle `IBinarySearchTreeIterable`
- Klasse `TreeTraversalType`

(a) *(4 Punkte)* Implementieren Sie die Knotenklasse `BinaryTreeNode`, die die Schnittstelle `IBinaryTreeNode` implementiert. Die Knotenklasse muss einen Standard-Konstruktor (d.h. Konstruktor ohne Parameter) enthalten.

(b) *(10 Punkte)* Implementieren Sie die Klasse `BinarySearchTree`, die die Schnittstelle `IBinarySearchTree` implementiert (ohne die bestehenden Methoden zu modifizieren).
Hinweis: Die Methode `remove` der Iteratoren wird nicht unterstützt und sollte immer `UnsupportedOperationException` werfen.