

Algorithmen und Datenstrukturen

Sommersemester 2009

4. Übungsblatt

Aufgabe 1: AVL-Bäume

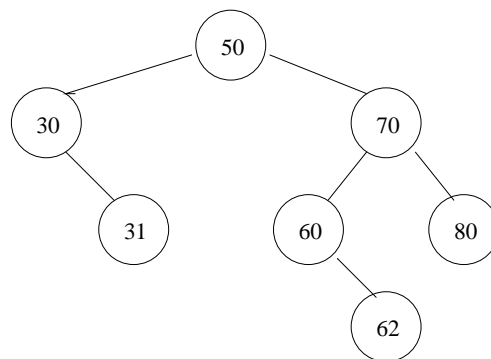


Abbildung 1: Ausgangsbaum für Aufgabenteil a)

- Fügen Sie in den Ausgangsbaum für a) die Knotenliste 20 , 18 , 17 , 61 ein. Zeichnen Sie dabei den AVL-Baum nach jedem Einfüge- und nach jedem Rotationsvorgang. Schreiben Sie die zugehörigen Fallbezeichnungen aus dem Vorlesungsalgorithmus daneben! (3 Punkte)
- Löschen Sie in dem Ausgangsbaum für b) die Knotenliste 59 , 70 , 30 , 21 , 19. Zeichnen Sie dabei den AVL-Baum nach jedem Lösch- und nach jedem Rotationsvorgang. Schreiben Sie die zugehörigen Fallbezeichnungen aus dem Vorlesungsalgorithmus daneben! (3 Punkte)

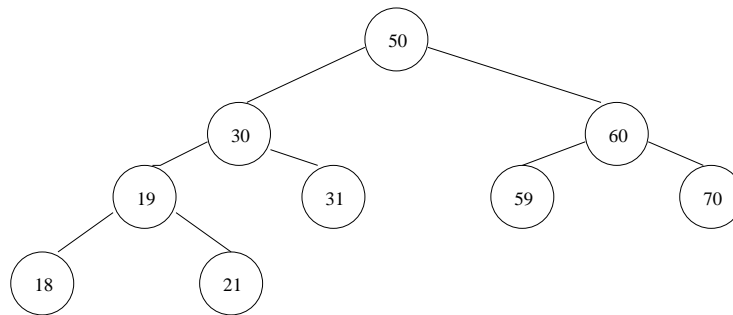


Abbildung 2: Ausgangsbaum für Aufgabenteil b)

Aufgabe 2: AVL-Bäume - Programmieren

In dieser Aufgabe sollen die in der Vorlesung besprochenen AVL-Bäume in Java programmiert werden. Auf der Übungsseite zur Vorlesung befindet sich hierzu fünf Java Dateien zum Herunterladen. Implementieren Sie das Java-Interface `AVLbaumInterface` mittels der Java-Klasse `AVLbaum`, die Sie bereits als Default-Implementierung (`AVLbaum.java`) vorfinden. Verwenden Sie für die Implementierung der Knoten die vorhandene Klasse `AVLNode`. `AVLNode` repräsentiert Knoten mit ganzen Zahlen als Inhalt. Damit Sie Ihren `AVLbaum` testen können, steht die Klasse `AVLDemo` bereits zur Verfügung. Falls Sie die `main`-Methode in `AVLDemo` aufrufen, können Sie Knoten mit ganzen Zahlen als Inhalt einfügen und löschen. `AVLDemo` greift auf Ihre Implementierung des AVL-Baums (`AVLbaum.java`) zurück! Ihre `AVLbaum`-Implementierung muss über einen Konstruktor mit leerer Parameterliste verfügen! Ausserdem ist beim Löschen eines Knotens mit zwei nichtleeren Unterbäumen stets das größte Element des linken Unterbaums als Ersatz zu verwenden!

- a) Implementieren Sie die Methoden `gibWurzel`, `gibLinks`, `gibRechts`, `gibVater`, `gibHoehe` und `gibBalance`. Die `Balance` soll nicht jeweils neu berechnet werden, sondern als Instanzvariable zur Verfügung stehen. (4 Punkte)
- b) Implementieren Sie die Methode `einFuegen` mittels geeigneter privater Hilfsmethoden, z.B. `ausbalancierenEinFuegen`, `rechtsRotation` und `linksRotation`. (10 Punkte)

Schicken Sie Ihre Lösung (`AVLbaum.java`) wie auf Übungsblatt 3 in den Hinweisen beschrieben an Ihren Übungsgruppenleiter.

Abgabetermin: Donnerstag, 07. Mai 2009, bis 11 Uhr Abgabekasten IFIS