

AI-3 AD — Praktikum #4

Aufgabenaufteilung:

Beide Teammitglieder haben einen eigenen Algorithmus zum Erstellen eines AVL Baums implementiert. Später wurden beide Lösungen zusammengeführt und die zusätzlich geforderten Logging Methoden gemeinsam implementiert.

Florian Kletz hat bearbeitet:

- s.o.

Micha Severin hat bearbeitet:

- s.o.

Quellenangaben:

1. JavaDocs

Begründung für Codeübernahme

1. Es wurde kein Code aus anderen Quellen übernommen.

Bearbeitungszeitraum:

Florian Kletz	Montag 10.06.2013 09:00-18:00
	Mittwoch 12.06.2013 09:00-18:00
Micha Severin	Dienstag 11.06.2013 14:00-17:00
	Donnerstag 13.06.2013 10:00-11:00
Gemeinsam	Freitag 14.06.2013 12:00-16:00
	Sonntag 16.06.2013 14:00-17:00

Aktueller Stand:

- Die Algorithmen wurden implementiert und getestet.
- Logging wurde implementiert.
- Einlesen von Zahlenfolgen wurde implementiert.

Probleme:

Wir wissen nicht, wofür die Postorder/Preorder Addition sinnvoll und nötig ist. Laut der gegebenen Berechnung könnte diese auch unabhängig von dem AVL-Baum berechnet werden.

Arbeitsweise des Algorithmus:

AVL-Bäume sind balancierte binäre Bäume. D.h. die Höhe der Teilbäume unterscheidet sich maximal um 1, falls der Unterschied höher ist, wird ein so genanntes Rotieren durchgeführt.

Ein AVL-Baum zeichnet sich dadurch aus, dass der Elternknoten immer größer als sein linker Kindknoten ist. Der rechte Kindknoten ist immer größer.

Beim Einfügen eines neuen Knoten wird nun von dem Wurzelknoten aus nach diesen Eigenschaften vorgegangen. Falls kein Wurzelknoten existiert, wird dieser neue Knoten automatisch zum Wurzelknoten. Sobald ein Kindknoten keinen weiteren Knoten mehr unter sich hat, wird der neue Knoten dort angefügt.

Der Einfügeprozess unterliegt den Gesetzen der Rotation. Ein AVL-Baum gilt als unbalanciert, sobald ein Knoten den Balancefaktor 2 aufweist. Die Regeln der Rotation nehmen wir als gegeben an und erklären diese hier nicht im Detail.

Rechtsrotation *	doppelte Rechtsrotation *	Problemfall *

* Linksrotationen analog

Der Balancefaktor eines Knoten berechnet sich nach folgender Formel:

$bal(x) = (\text{Höhe des rechten Unterbaumes von } x) - (\text{Höhe des linken Unterbaumes von } x)$

Es gibt drei mögliche Fallunterscheidungen: Hat der zu löschende Knoten keine Kinder, so wird der Knoten einfach nur gelöscht. Hat der Knoten genau ein Kind, so wird der gelöschte Knoten mit diesem ersetzt. Hat er zwei Kinder so gibt es zwei mögliche Vorgehensweisen. Von einem Knoten X wird der Knoten Y an die Position von X gesetzt. Der Knoten Y ist der maximale Knoten des linken Teilsbaums von X. Die zweite Vorgehensweise ist, dass man den minimalen Knoten Y des rechten Teilbaums von X nimmt.