Datenstrukturen, Algorithmen und Programmierung 2

Amin Coja-Oghlan

July 14, 2022

Lehrstuhl Informatik 2
Fakultät für Informatik

AVL-Bäume

Worum geht es?

- AVL-Bäume, benannt nach ihren Erfindern Adelson-Velsky und Landis sind eine Alternative zu rot-schwarz-Bäumen
- alle Operationen benötigen Zeit $O(\log n)$
- wiederum müssen wir die Operationen “einfügen” und “entfernen” anpassen

AVL-Bäume

Definition

Ein AVL-Baum ist ein gewurzelter binärer Suchbaum, in dem sich für jeden Knoten v die Höhe des linken und des rechten Teilbaums um höchstens eins unterscheidet.

- folglich haben AVL-Bäume Höhe $O(\log n)$
- somit haben die Operationen `Search`, `Minimum`, `Maximum`, `Successor`, `Predecessor` Laufzeit $O(\log n)$

AVL-Bäume

Höhenbalance

- um die AVL-Definition zu erhalten, fügen wir jedem Knoten des Baumes ein weiteres Datenelement hinzu
- dieser zusätzliche Eintrag ist die *Höhe* des Teilbaums, der an diesem Knoten "hängt"
- genauer gesagt genügt die Balance

$$\beta(v) = \text{Höhe des rechten Teilbaums von } v - \text{Höhe des linken Teilbaums von } v.$$

AVL-Bäume

Höhenbalance

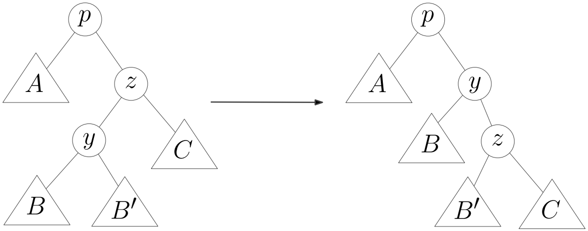
- wir fügen zunächst einen neuen Knoten z wie in einen gewöhnlichen binären Suchbaum ein
- dadurch kann allerdings die AVL-Eigenschaft zerstört werden
- um sie wiederherzustellen, führen wir geeignete Rotationen
- diese werden rekursiv von unten nach oben auf dem Pfad zur Wurzel ausgeführt
- dabei bezeichnet β auf die Balance *vor* dem Einfügen des neuen Knotens z

AVL-Bäume

Balancieren nach Einfügen

1. falls z die Wurzel ist, halte; sonst sei p der Elternknoten von z.
2. falls z das rechte Kind von p ist
3.   falls $\beta(p) > 0$
4.   falls $\beta(z) < 0$
5.   Rechtsrotation um z, anschließend Linksrotation um p
6.   sonst Linksrotation um p
7.   sonst
8.   falls $\beta(p) < 0$, setze $\beta(p) = 0$ und halte
9.   sonst setze $\beta(p) = 1$
10. *sonst verfahre analog wie oben mit links/rechts vertauscht*

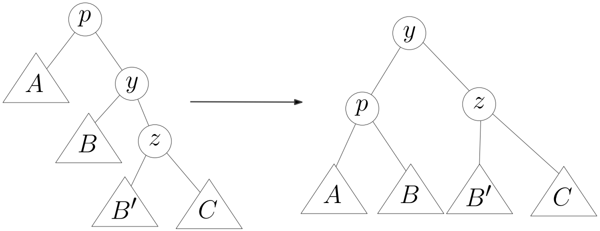
AVL-Bäume



$\beta(p) > 0$ und $\beta(z) < 0$

- Rechtsrotation um z
- dann Linksrotation um p

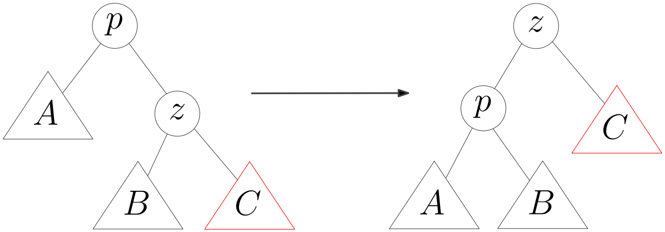
AVL-Bäume



$\beta(p) > 0$ **und** $\beta(z) < 0$

- Rechtsrotation um z
- dann Linksrotation um p

AVL-Bäume



$\beta(p) > 0$ und $\beta(z) \geq 0$

- Linksrotation um p

AVL-Bäume

Balancieren nach Löschen

1. falls z die Wurzel ist, halte.
2. sei p der Elternknoten von z und y der Schwesterkonten von z (bzw. $y = \emptyset$)
3. falls z das linke Kind von p ist
4.   falls $\beta(p) > 0$
5.   falls $\beta(y) < 0$
6.   Rechtsrotation um y, anschließend Linksrotation um p
7.   sonst Linksrotation um p
8.   sonst
9.   falls $\beta(p) = 0$, setze $\beta(p) = 1$ und halte
10.   sonst setze $\beta(p) = 0$
11. *sonst verfahre analog wie oben mit links/rechts vertauscht*

AVL-Bäume

Zusammenfassung

- AVL-Bäume sind eine Alternative zu rot-schwarz-Bäumen
- Balancieroperation verwendet ausschließlich Rotationen