ADS: Algorithmen und Datenstrukturen

Teil IX

Peter F. Stadler & Konstantin Klemm

Bioinformatics Group, Dept. of Computer Science & Interdisciplinary Center for Bioinformatics, **University of Leipzig**

06. Januar 2010

Wiederholung: l-balancierter Binärbaum

Seien $B_l(x)$ und $B_r(x)$ die linken und rechten Unterbäume eines Knotens x. Weiterhin sei $h(B)$ die Höhe eines Baumes B.

Definition: Ein l-balancierter Binärbaum ist entweder leer oder es ist ein Baum, bei dem für jeden Knoten x gilt:

$$|h(B_l(x)) - h(B_r(x))| \leq l \ .$$

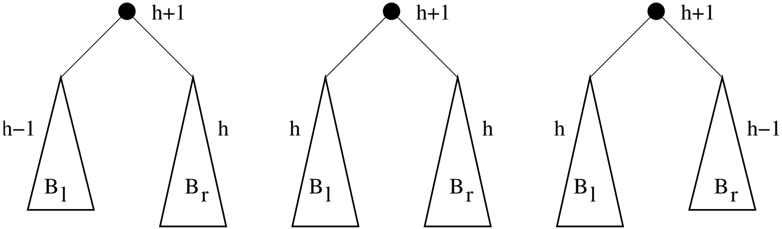
l läßt sich als Maß für die zulässige Entartung im Vergleich zur ausgeglichenen Baumstruktur auffassen.

AVL-Baum

- nach russischen Mathematikern **A**delson-**V**elski und **L**andis
- Definition: Ein 1-balancierter Binärbaum heißt AVL-Baum.
- Balancierungskriterium

$$|h(B_l(x)) - h(B_r(x))| \leq 1 \ .$$

- Konstruktionsprinzip:



AVL-Baum: Wartungsalgorithmen

Wann und wo ist das AVL-Kriterium beim Einfügen verletzt?

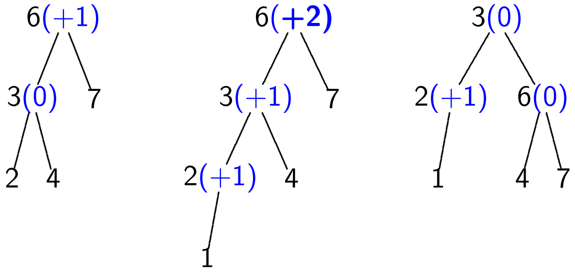
- Es kann sich nur die Höhe von solchen Unterbäumen verändert haben, deren Wurzeln auf dem Suchpfad von der Wurzel des Baumes zum neu eingefügten Blatt liegen.
- Reorganisationsoperationen lassen sich lokal begrenzen. Höchstens h Knoten sind betroffen.
- Der Balancierungsfaktor $\mathrm{BF}(x)$ eines Knotens x ergibt sich zu

$$\mathrm{BF}(x) = h(B_l(x)) - h(B_r(x)) \ .$$

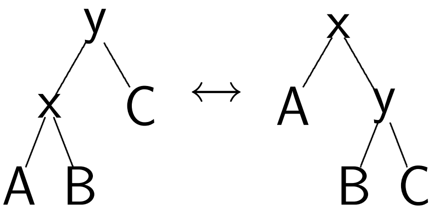
$=$ Höhendifferenz der Unterbäume von x.

Beispiel: Einfügen und Einfachrotation

AVL-Baum 1 Einfügen Rechtsrotation



Einfachrotationen



x,y = Knoten
A,B,C = Teilbäume

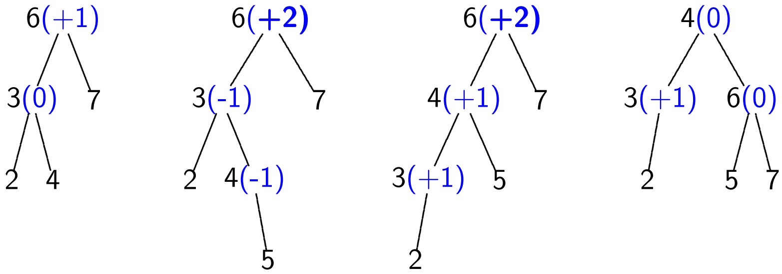
Beispiel: Einfügen und Doppelrotation

AVL-Baum

5 Einfügen

Linksrotation
$3 \curvearrowleft 4$

Rechtsrotation
$4 \curvearrowright 6$



AVL-Rebalancierung

Jeweils nach Einfügen oder Löschen eines Knotens i, für alle Knoten y auf dem Pfad vom Vater von i bis zur Wurzel:

- Berechne BF(y).
- Falls $\mathrm{BF}(y) \geq +2$: Betrachte linken Tochterknoten x von y. Bei $\mathrm{BF}(x) \geq 0$, rotiere rechts um y. Sonst ($\mathrm{BF}(x) < 0$), rotiere zuerst links um x, dann rechts um y.
- Falls $\mathrm{BF}(y) \leq -2$: Betrachte rechten Tochterknoten x von y. Bei $\mathrm{BF}(x) \leq 0$, rotiere links um y. Sonst ($\mathrm{BF}(x) > 0$), rotiere zuerst rechts um x, dann links um y.

Nach *Einfügen* eines Knotens ist maximal eine Einfach- oder Doppelrotation nötig. Der betroffenene Teilbaum hat dieselbe Höhe wie vor dem Einfügen, so dass die Balancierungsfaktoren oberhalb nicht neu zu berechnen sind.

Höhe von AVL-Bäumen I

Sei ν_h die Anzahl Knoten, die ein AVL-Baum der Höhe h mindestens hat.

- $\nu_0 = 0$ (leerer Baum), $\nu_1 = 1$ (nur Wurzel)
- Rekursion for $h \geq 2$

$$\nu_h = \nu_{h-1} + \nu_{h-2} + 1$$

- denn: AVL-Baum der Höhe h mit geringster Anzahl Knoten hat mindestens einen Unterbaum der Höhe $h - 1$, der andere Unterbaum hat aber Höhe $h - 2$ (sonst ist Knotenzahl nicht minimal wegen $\nu_{h-2} < \nu_{h-1}$).
- Zur Vereinfachung Abschätzung von unten:

$$\nu_h \geq \nu_{h-1} + \nu_{h-2}$$

$=$ Rekursions(un)gleichung für Fibonacci-Zahlen.

Höhe von AVL-Bäumen II

Somit ist für $h \geq 0$

$$\nu_h \geq F_h$$

mit der h-ten Fibonacci-Zahl (vgl. 1. Vorlesung)

$$F_h = (A^h - B^h)/\sqrt{5}\ ,$$

wobei $A = (1+\sqrt{5})/2$ und $B = (1-\sqrt{5})/2$. Da $|B| < 1$, ist B^h durch 1 beschränkt und wir können weiter abschätzen

$$\nu_h \geq (A^h + 1)/\sqrt{5}\ .$$

Also

$$h \leq \frac{\log_2(\sqrt{5}\nu - 1)}{\log_2 A} < \frac{\log_2 \nu}{\log_2 A}$$

für $\nu \geq 2$.

Höhe von AVL-Bäumen III

Endergebnis für AVL-Bäume der Höhe h mit n Knoten:

$$h < 1.44 \log_2 n$$

Vergleich mit vollständigen Bäumen:

$$h = \log_2(n+1)$$

$\Rightarrow$ AVL-Bäume im ungünstigsten Fall um Faktor 1.44 höher als vollständige Bäume mit derselben Knotenzahl.

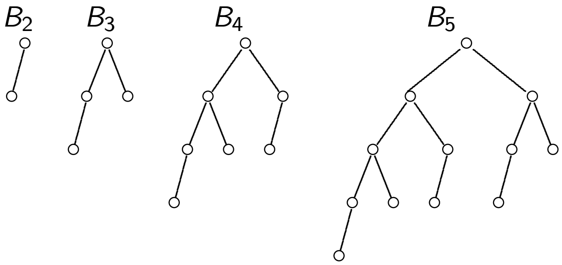
Wie sehen nun die am wenigsten balancierten AVL-Bäume aus?

Fibonacci-Bäume

"Maximal unbalancierte" AVL-Bäume.

- Zu jedem $h \geq 0$ gibt es genau einen Fibonacci-Baum B_h mit Höhe h.
- B_0 ist der leere Baum, B_1 ist ein einzelner Knoten
- für $h \geq 2$ ist der Fibonacci-Baum der Höhe h mit Wurzel x.

$$B_h = \mathrm{BUILD}(B_{h-1}, x, B_{h-2}) \ .$$



Gewichtsbalancierte Suchbäume I

Gewichtsbalancierte oder BB-Bäume (bounded balance)

Zulässige Abweichung der Struktur vom ausgeglichenen Binärbaum wird als Differenz zwischen der Anzahl der Knoten im rechten und linken Unterbaum festgelegt.

Definition: Sei B ein binärer Suchbaum mit $n > 0$ Knoten, von denen sich n_l Knoten im linkem Unterbaum B_l befinden.

- $\rho(B) = \frac{n_l+1}{n+1}$ heißt die Wurzelbalance von B.
- Ein Baum B heißt gewichtsbalanciert, BB(α), oder von beschränkter Balance α, wenn für jeden Unterbaum B' von B gilt:

$$\alpha \leq \rho(B') \leq 1 - \alpha$$

Nievergelt and Reingold (1972), `http://portal.acm.org/citation.cfm?id=804906`

Gewichtsbalancierte Suchbäume II

Wahl des Parameters α:

- $\alpha = 1/2$: Balancierungskriterium akzeptiert nur vollständige Binärbäume
- $\alpha < 1/2$: Strukturbeschränkung wird zunehmend gelockert
- $\alpha = 0$: keine Beschränkung der Baumstruktur

Wartung

- Einsatz derselben Rotationstypen wie beim AVL-Baum
- Rebalancierung ist gewährleistet durch eine Wahl von

$$\alpha \leq 1 - \sqrt{\frac{1}{2}}$$

Kosten für Suche und Aktualisierung: $O(\log n)$

Positionssuche mit balancierten Bäumen I

Balancierte Suchbäume

- sind linearen Listen in fast allen Grundoperationen überlegen
- Lösung des Auswahlproblems bzw. Positionssuche (Suche nach k-tem Element der Sortierreihenfolge) kann jedoch noch verbessert werden

Definition: Der *Rang* eines Knotens ist die um 1 erhöhte Anzahl der Knoten seines linken Unterbaums. (Blattknoten haben Rang 1)

Positionssuche mit balancierten Bäumen II

Rangzahlen erlauben Bestimmung eines direkten Suchpfads im Baum für Positionssuche nach dem k-ten Element.

- Position $p := k$; beginne Suche am Wurzelknoten
- Wenn Rang r eines Knotens $= p$, gilt: Element gefunden
- falls $r > p$: Suche im linken Unterbaum des Knotens weiter
- falls $r < p$: Ersetze p := p - r und setze Suche im rechten UB fort.

Die Wartungsoperationen erfordern etwas mehr Aufwand: Nach Einfügen / Löschen eines Knotens müssen die Rangwerte auf dem kompletten Suchpfad angepasst werden.