

# 18. AVL-Bäume

---

Balancierte Bäume [Ottman/Widmayer, Kap. 5.2-5.2.1, Cormen et al, Kap. Problem 13-3]

# Hintergrund

- Suchbaum: Suchen, Einfügen und Entfernen eines Schlüssels im Mittel in  $\mathcal{O}(\log n)$  Schritten (bei  $n$  Schlüsseln im Baum)
- Schlechtester Fall jedoch:  $\Theta(n)$  (degenerierter Baum)

# Hintergrund

- Suchbaum: Suchen, Einfügen und Entfernen eines Schlüssels im Mittel in  $\mathcal{O}(\log n)$  Schritten (bei  $n$  Schlüsseln im Baum)
- Schlechtester Fall jedoch:  $\Theta(n)$  (degenerierter Baum)

**Ziel:** Verhindern der Degenerierung, durch Balancieren des Baumes nach jeder Update-Operation.

*Balancierung:* Garantiere, dass ein Baum mit  $n$  Knoten stets eine Höhe von  $\mathcal{O}(\log n)$  hat.

# Hintergrund

- Suchbaum: Suchen, Einfügen und Entfernen eines Schlüssels im Mittel in  $\mathcal{O}(\log n)$  Schritten (bei  $n$  Schlüsseln im Baum)
- Schlechtester Fall jedoch:  $\Theta(n)$  (degenerierter Baum)

**Ziel:** Verhindern der Degenerierung, durch Balancieren des Baumes nach jeder Update-Operation.

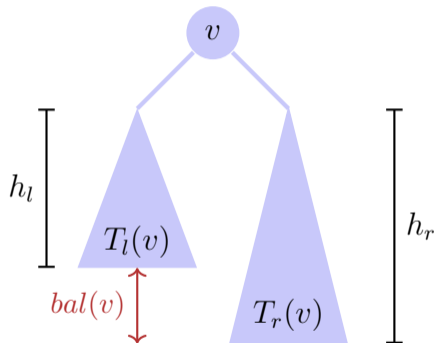
*Balancierung:* Garantiere, dass ein Baum mit  $n$  Knoten stets eine Höhe von  $\mathcal{O}(\log n)$  hat.

**Adelson-Velski und Landis (1962): AVL-Bäume**

# Balance eines Knotens

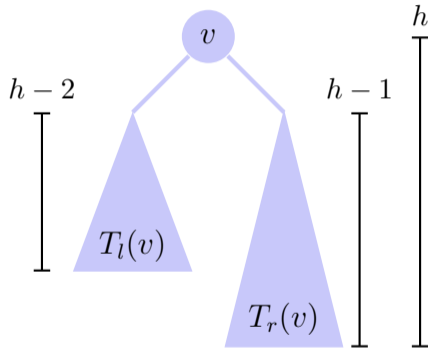
Die *Balance* eines Knotens  $v$  ist definiert als die Höhendifferenz seiner beiden Teilbäume  $T_l(v)$  und  $T_r(v)$

$$\text{bal}(v) := h(T_r(v)) - h(T_l(v))$$

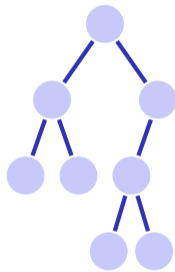
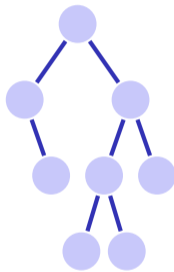
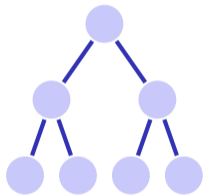


# AVL-Bedingung

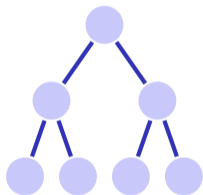
*AVL-Bedingung*: für jeden Knoten  $v$  eines Baumes gilt  $\text{bal}(v) \in \{-1, 0, 1\}$



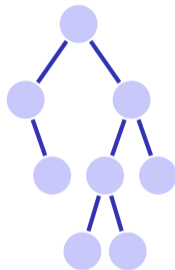
# (Gegen-)Beispiele



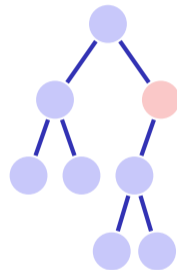
# (Gegen-)Beispiele



AVL-Baum der Höhe 3



AVL-Baum der Höhe 4



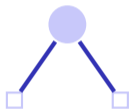
Kein AVL-Baum



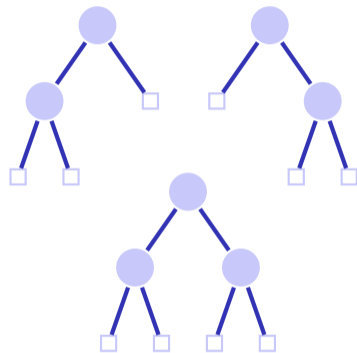
# Anzahl Blätter

- 1. Beobachtung: Ein Binärbaum mit  $n$  Schlüsseln hat genau  $n + 1$  Blätter. Einfaches Induktionsargument.
  - Der Binärbaum mit  $n = 0$  Schlüsseln hat  $m = 1$  Blätter
  - Wird ein Schlüssel (Knoten) hinzugefügt ( $n \rightarrow n + 1$ ), so ersetzt er ein Blatt und fügt zwei Blätter hinzu ( $m \rightarrow m - 1 + 2 = m + 1$ ).
- 2. Beobachtung: Untere Grenze für Anzahl Blätter eines Binärbaums zu gegebener Höhe erlaubt Abschätzung der maximalen Höhe eines Binärbaums zu gegebener Anzahl Schlüssel.

# Untere Grenze Blätter



AVL-Baum der Höhe 1 hat  
 $N(1) := 2$  Blätter



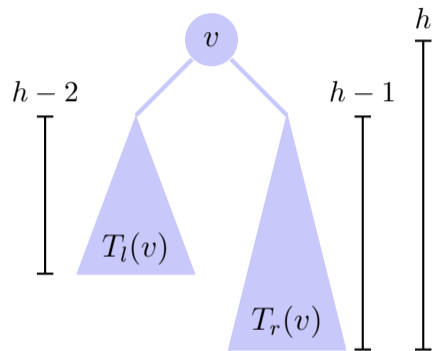
AVL-Baum der Höhe 2 hat  
mindestens  $N(2) := 3$  Blätter

# Untere Grenze Blätter für $h > 2$ in AVL-Bäumen

- Höhe eines Teilbaums  $\geq h - 1$ .
- Höhe des anderen Teilbaums  $\geq h - 2$ .

Minimale Anzahl Blätter  $N(h)$  ist

$$N(h) = N(h - 1) + N(h - 2)$$



Insgesamt gilt  $N(h) = F_{h+2}$  mit **Fibonacci-Zahlen**  $F_0 := 0$ ,  $F_1 := 1$ ,  
 $F_n := F_{n-1} + F_{n-2}$  für  $n > 1$ .

# Fibonacci Zahlen, geschlossene Form

Es gilt<sup>22</sup>

$$F_i = \frac{1}{\sqrt{5}}(\phi^i - \hat{\phi}^i)$$

mit den Wurzeln  $\phi, \hat{\phi}$  der Gleichung vom goldenen Schnitt  $x^2 - x - 1 = 0$ :

$$\phi = \frac{1 + \sqrt{5}}{2} \approx 1.618$$

$$\hat{\phi} = \frac{1 - \sqrt{5}}{2} \approx -0.618$$

---

<sup>22</sup>Herleitung mit Erzeugendenfunktionen (Potenzreihen) im Anhang

Da  $|\hat{\phi}| < 1$ , gilt insgesamt

$$N(h) \in \Theta\left(\left(\frac{1 + \sqrt{5}}{2}\right)^h\right) \subseteq \Omega(1.618^h)$$

und somit

$$N(h) \geq c \cdot 1.618^h \quad \Rightarrow \quad h \leq 1.44 \log_2 n + c'$$

- D.h. ein AVL-Baum hat die gewünschte Höhe von  $\mathcal{O}(\log n)$
- und ist asymptotisch nicht mehr als 44% höher als ein perfekt balancierter Baum (Höhe  $\lceil \log_2 n + 1 \rceil$ )

# Einfügen und Balancieren

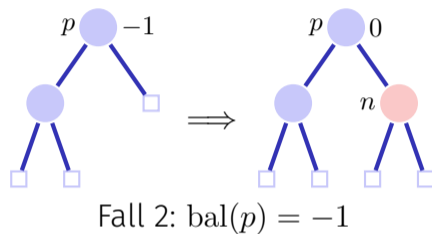
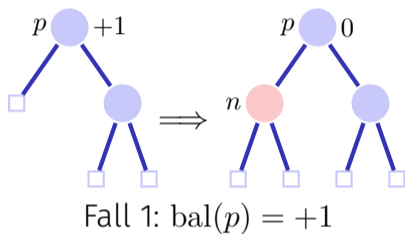
Balancieren:

- Einfügen verletzt möglicherweise AVL-Bedingung → Balancieren
- Dafür wird die Balance an jedem Knoten gespeichert

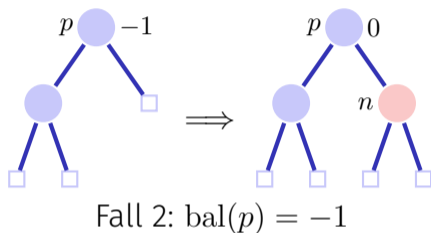
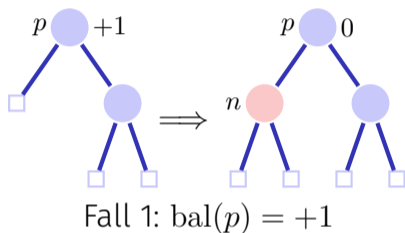
Einfügen:

- Neuen Knoten  $n$  einfügen, wie beim Suchbaum
- Dann Prüfen und ggf. Wiederherstellen der Balance für alle Knoten von  $n$  aufwärts bis zur Wurzel

# Balance am Einfügeort



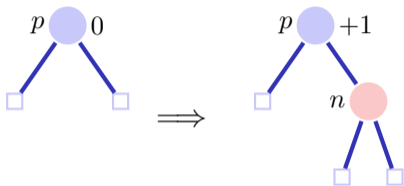
# Balance am Einfügeort



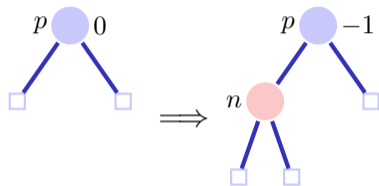
Direkt fertig in beiden Fällen, denn die Höhe des Teilbaums  $p$  hat sich nicht verändert. Balance des Elternknoten daher ebenfalls unverändert.



# Balance am Einfügeort

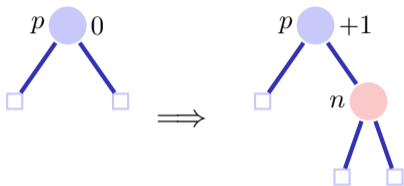


Fall 3.1:  $\text{bal}(p) = 0$  rechts

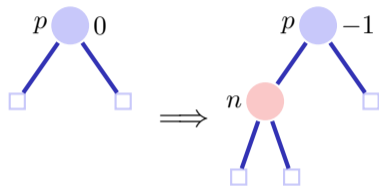


Fall 3.2:  $\text{bal}(p) = 0$ , links

# Balance am Einfügeort



Fall 3.1:  $\text{bal}(p) = 0$  rechts



Fall 3.2:  $\text{bal}(p) = 0$ , links

In beiden Fällen noch nicht fertig, da Elternknoten nun möglicherweise nicht mehr balanciert  $\rightarrow$  Aufruf der Funktion **upin(p)** (upwards + insert)

# upin(p): Rekursive Aufrufbedingung

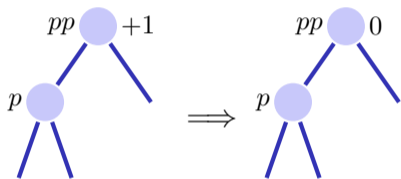
Bei jedem Aufruf **upin(p)** muss gelten, dass

- der Teilbaum  $p$  gewachsen ist und dadurch
- $\text{bal}(p)$  von 0 auf  $\in \{-1, +1\}$  geändert wurde.

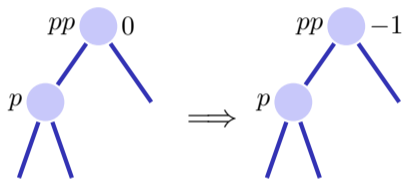
Denn nur in dieser Situation kann das neu entstandene Ungleichgewicht von  $p$  ( $\text{bal}(p) \neq 0$ ) einen Einfluss auf die Baumstruktur darüber haben.

# upin(p)

Annahme:  $p$  ist linker Sohn von  $pp^{23}$



Fall 1:  $\text{bal}(pp) = +1$ , fertig.



Fall 2:  $\text{bal}(pp) = 0$ , **upin(pp)**

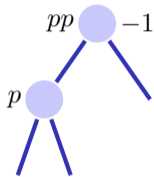
In beiden Fällen gilt nach der Operation die AVL-Bedingung für den Teilbaum ab  $pp$

---

<sup>23</sup>Ist  $p$  rechter Sohn: symmetrische Fälle unter Vertauschung von  $+1$  und  $-1$

# upin(p)

Annahme:  $p$  ist linker Sohn von  $pp$



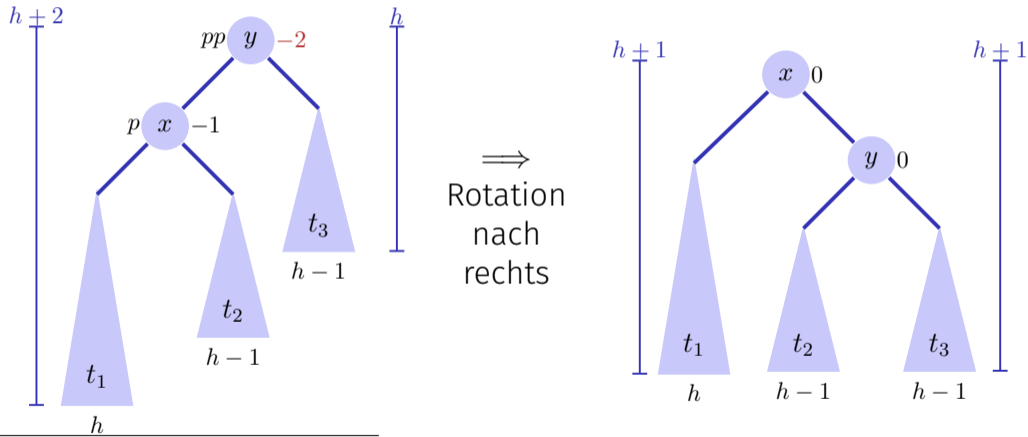
Fall 3:  $\text{bal}(pp) = -1,$

Dieser Fall ist problematisch: das Hinzufügen von  $n$  im Teilbaum ab  $pp$  hat die AVL-Bedingung verletzt. Rebalancieren!

Zwei Fälle  $\text{bal}(p) = -1, \text{bal}(p) = +1$

# Rotationen

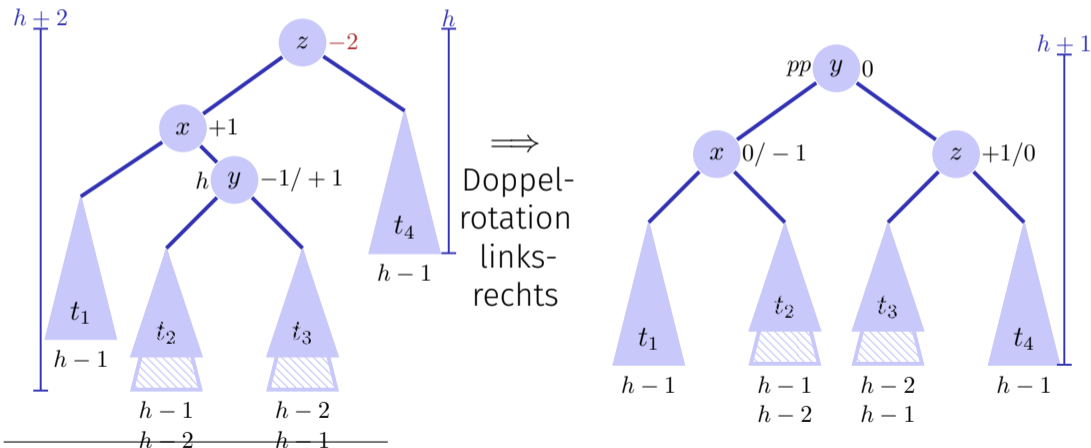
Fall 1.1  $\text{bal}(p) = -1$ .<sup>24</sup>



<sup>24</sup> $p$  rechter Sohn  $\Rightarrow \text{bal}(pp) = \text{bal}(p) = +1$ , Linksrotation

# Rotationen

Fall 1.2  $\text{bal}(p) = +1$ .<sup>25</sup>



<sup>25</sup> $p$  rechter Sohn  $\Rightarrow \text{bal}(pp) = +1, \text{bal}(p) = -1$ , Doppelrotation rechts links

- Höhe des Baumes:  $\mathcal{O}(\log n)$ .
- Einfügen wie beim binären Suchbaum.
- Balancieren durch Rekursion vom Knoten zur Wurzel (beim rekursiven Aufstieg). Maximale Pfadlänge  $\mathcal{O}(\log n)$ .

Das Einfügen im AVL-Baum hat Laufzeitkosten von  $\mathcal{O}(\log n)$ .



Entfernen eines Knotens aus einem AVL-Baum zieht ebenfalls Rotationen nach sich, ist aber noch etwas komplexer – und nicht prüfungsrelevant. Bei Interesse finden Sie mehr Informationen im Handout.

# Zusammenfassung

- AVL-Bäume haben asymptotische Laufzeit von  $\mathcal{O}(\log n)$  (schlechtester Fall) für das Suchen, Einfügen und Löschen von Schlüssel
- Einfügen und Löschen ist verhältnismässig aufwändig. Bei kleinen Bäumen (Schlüsselmenge) überwiegen die Rotationskosten den Gewinn durch  $\mathcal{O}(\log n)$  Höhe.
- Diverse andere balancierte Bäume existieren: Red-Black tree (`std::map` in C++), B-tree (`std::collections::BTreeMap` in Rust), Splay tree; Treap (mit hoher Wahrscheinlichkeit balanciert)

## 18.6 Anhang

---

Herleitung einiger mathematischen Formeln

# Fibonacci Zahlen, Induktiver Beweis

$$F_i \stackrel{!}{=} \frac{1}{\sqrt{5}}(\phi^i - \hat{\phi}^i) \quad [*] \quad \left(\phi = \frac{1+\sqrt{5}}{2}, \hat{\phi} = \frac{1-\sqrt{5}}{2}\right).$$

1. Klar für  $i = 0, i = 1$ .
2. Sei  $i > 2$  und Behauptung  $[*]$  wahr für alle  $F_j, j < i$ .

$$\begin{aligned} F_i &\stackrel{\text{def}}{=} F_{i-1} + F_{i-2} \stackrel{[*]}{=} \frac{1}{\sqrt{5}}(\phi^{i-1} - \hat{\phi}^{i-1}) + \frac{1}{\sqrt{5}}(\phi^{i-2} - \hat{\phi}^{i-2}) \\ &= \frac{1}{\sqrt{5}}(\phi^{i-1} + \phi^{i-2}) - \frac{1}{\sqrt{5}}(\hat{\phi}^{i-1} + \hat{\phi}^{i-2}) = \frac{1}{\sqrt{5}}\phi^{i-2}(\phi + 1) - \frac{1}{\sqrt{5}}\hat{\phi}^{i-2}(\hat{\phi} + 1) \end{aligned}$$

$(\phi, \hat{\phi}$  erfüllen  $x + 1 = x^2$ )

$$= \frac{1}{\sqrt{5}}\phi^{i-2}(\phi^2) - \frac{1}{\sqrt{5}}\hat{\phi}^{i-2}(\hat{\phi}^2) = \frac{1}{\sqrt{5}}(\phi^i - \hat{\phi}^i).$$

# [Fibonacci Zahlen: geschlossene Form]

Geschlossene Form der Fibonacci Zahlen: Berechnung über erzeugende Funktionen:

1. Potenzreihenansatz

$$f(x) := \sum_{i=0}^{\infty} F_i \cdot x^i$$

# [Fibonacci Zahlen: geschlossene Form]

2. Für Fibonacci Zahlen gilt  $F_0 = 0$ ,  $F_1 = 1$ ,  $F_i = F_{i-1} + F_{i-2} \forall i > 1$ . Daher:

$$\begin{aligned} f(x) &= x + \sum_{i=2}^{\infty} F_i \cdot x^i = x + \sum_{i=2}^{\infty} F_{i-1} \cdot x^i + \sum_{i=2}^{\infty} F_{i-2} \cdot x^i \\ &= x + x \sum_{i=2}^{\infty} F_{i-1} \cdot x^{i-1} + x^2 \sum_{i=2}^{\infty} F_{i-2} \cdot x^{i-2} \\ &= x + x \sum_{i=0}^{\infty} F_i \cdot x^i + x^2 \sum_{i=0}^{\infty} F_i \cdot x^i \\ &= x + x \cdot f(x) + x^2 \cdot f(x). \end{aligned}$$

# [Fibonacci Zahlen: geschlossene Form]

3. Damit:

$$f(x) \cdot (1 - x - x^2) = x.$$
$$\Leftrightarrow f(x) = \frac{x}{1 - x - x^2} = -\frac{x}{x^2 + x - 1}$$

Mit den Wurzeln  $-\phi$  und  $-\hat{\phi}$  von  $x^2 + x - 1$ ,

$$\phi = \frac{1 + \sqrt{5}}{2} \approx 1.6, \quad \hat{\phi} = \frac{1 - \sqrt{5}}{2} \approx -0.6.$$

gilt  $\phi \cdot \hat{\phi} = -1$  und somit

$$f(x) = -\frac{x}{(x + \phi) \cdot (x + \hat{\phi})} = \frac{x}{(1 - \phi x) \cdot (1 - \hat{\phi} x)}$$

# [Fibonacci Zahlen: geschlossene Form]

4. Es gilt:

$$(1 - \hat{\phi}x) - (1 - \phi x) = \sqrt{5} \cdot x.$$

Damit:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{5}} \frac{(1 - \hat{\phi}x) - (1 - \phi x)}{(1 - \phi x) \cdot (1 - \hat{\phi}x)} \\ &= \frac{1}{\sqrt{5}} \left( \frac{1}{1 - \phi x} - \frac{1}{1 - \hat{\phi}x} \right) \end{aligned}$$



# [Fibonacci Zahlen: geschlossene Form]

5. Potenzreihenentwicklung von  $g_a(x) = \frac{1}{1-a \cdot x}$  ( $a \in \mathbb{R}$ ):

$$\frac{1}{1-a \cdot x} = \sum_{i=0}^{\infty} a^i \cdot x^i.$$

Sieht man mit Taylor-Entwicklung von  $g_a(x)$  um  $x = 0$  oder so: Sei  $\sum_{i=0}^{\infty} G_i \cdot x^i$  eine Potenzreihenentwicklung von  $g$ . Mit der Identität  $g_a(x)(1 - a \cdot x) = 1$  gilt für alle  $x$  (im Konvergenzradius)

$$1 = \sum_{i=0}^{\infty} G_i \cdot x^i - a \cdot \sum_{i=0}^{\infty} G_i \cdot x^{i+1} = G_0 + \sum_{i=1}^{\infty} (G_i - a \cdot G_{i-1}) \cdot x^i$$

Für  $x = 0$  folgt  $G_0 = 1$  und für  $x \neq 0$  folgt dann  $G_i = a \cdot G_{i-1} \Rightarrow G_i = a^i$ .

# [Fibonacci Zahlen: geschlossene Form]

6. Einsetzen der Potenzreihenentwicklung:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{5}} \left( \frac{1}{1 - \phi x} - \frac{1}{1 - \hat{\phi} x} \right) = \frac{1}{\sqrt{5}} \left( \sum_{i=0}^{\infty} \phi^i x^i - \sum_{i=0}^{\infty} \hat{\phi}^i x^i \right) \\ &= \sum_{i=0}^{\infty} \frac{1}{\sqrt{5}} (\phi^i - \hat{\phi}^i) x^i \end{aligned}$$

Koeffizientenvergleich mit  $f(x) = \sum_{i=0}^{\infty} F_i \cdot x^i$  liefert

$$F_i = \frac{1}{\sqrt{5}} (\phi^i - \hat{\phi}^i).$$