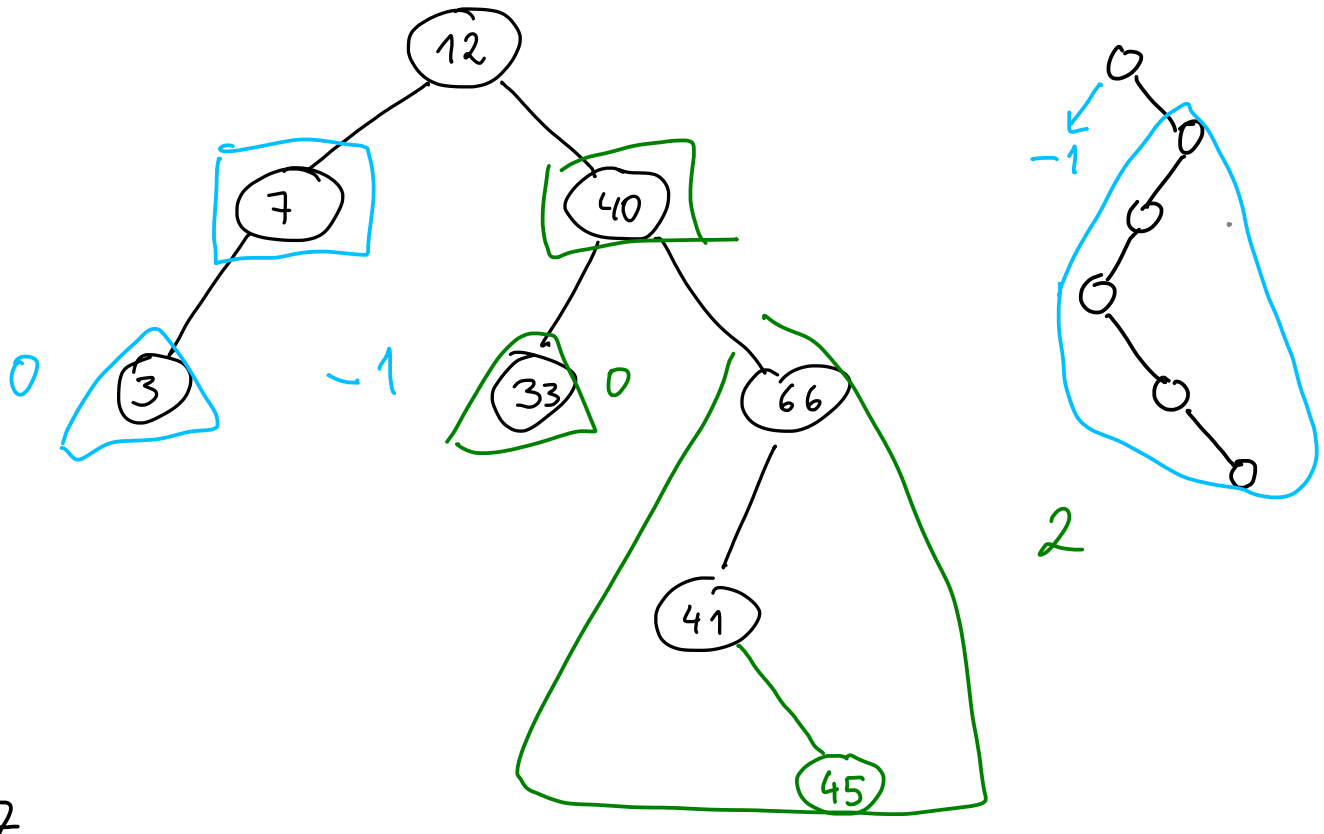
Höhenbalancierte Bäume: AVL-Bäume

In jedem Knoten gilt:

Die Höhen des linken und des rechten Teilbaums unterscheiden sich höchstens um 1.



SATZ.

Die Höhe h eines AVL-Baums mit n Knoten ist $O(\log n)$.

Adelson-Velski und Landis (1962)

Lemma:

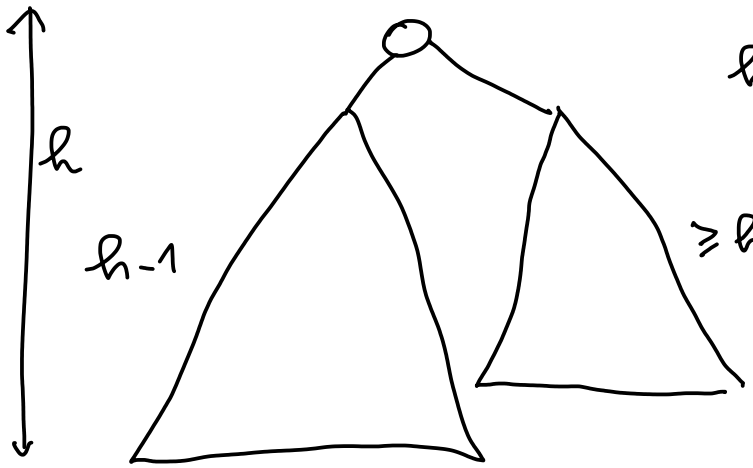
Ein AVL-Baum der Höhe h enthält $\geq 2^{h/2}$ Knoten.

$\Rightarrow$ Satz: $n \geq 2^{h/2}$ $\quad \log_2 n \geq h/2$ $\quad h \leq 2 \cdot \log_2 n = O(\log n)$ ✓

Beweis des Lemmas durch vollständige Induktion nach h.

$h=0$ $n \geq 1 \geq 2^0 = 1$ ✓

$h=1$ $n \geq 2$: $2 \geq 2^{1/2} = \sqrt{2}$ ✓



I.V. I.V.

$$h \geq 2: \quad n \geq 1 + 2^{\frac{h-1}{2}} + 2^{\frac{h-2}{2}}$$

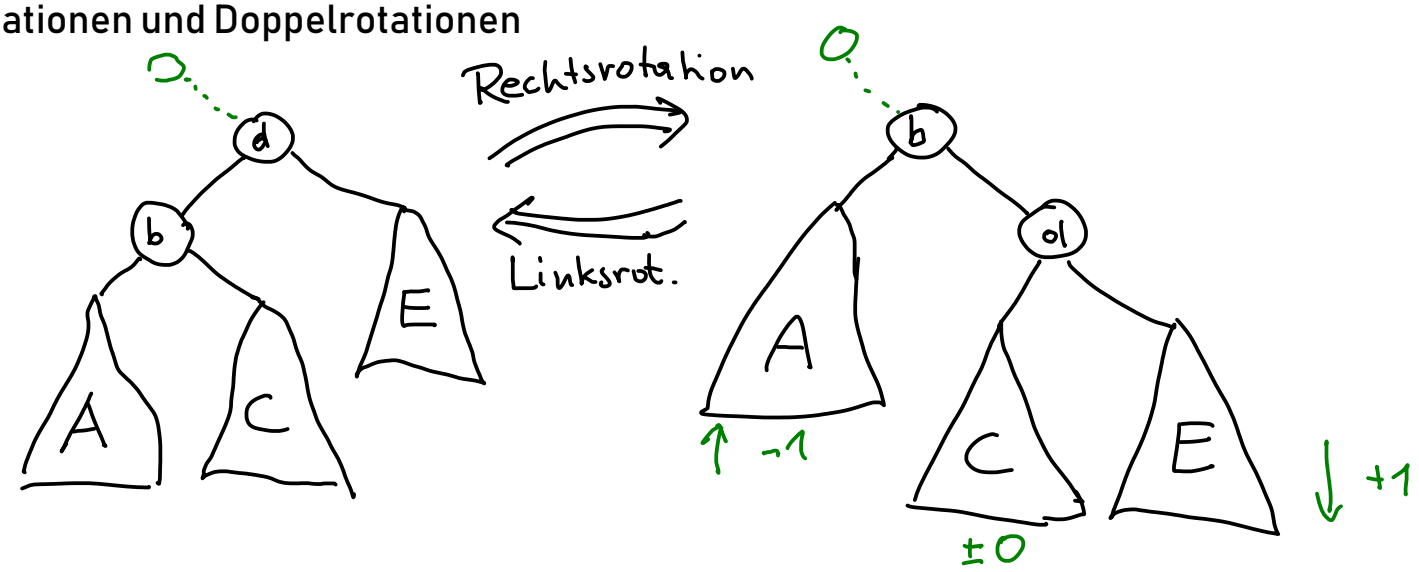
$$\geq 2^{\frac{h-2}{2}} + 2^{\frac{h-2}{2}} =$$

$$= 2 \cdot 2^{\frac{h-2}{2}} = 2^{\frac{h-2}{2}+1} = 2^{h/2}$$

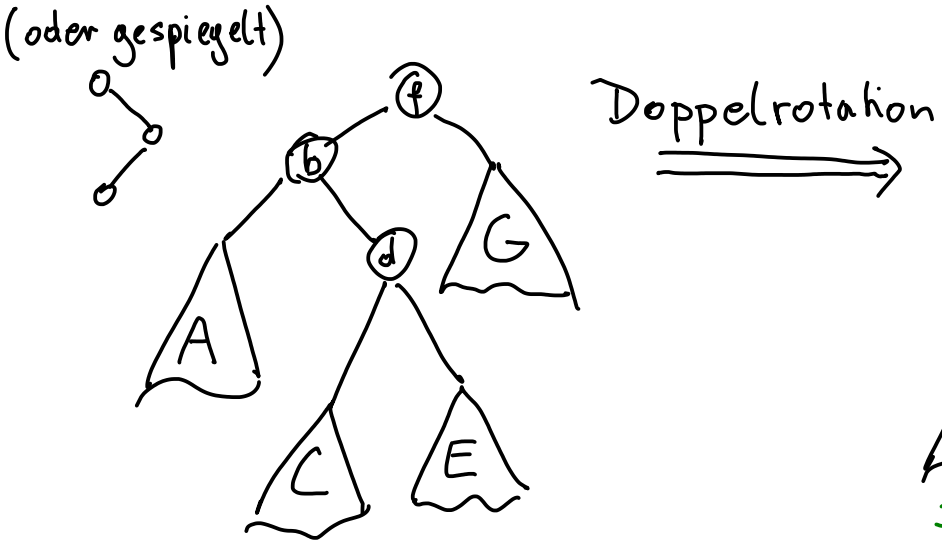
□

$h=2$: $n \geq 4$ $h=3$: $n \geq$

Rotationen und Doppelrotationen



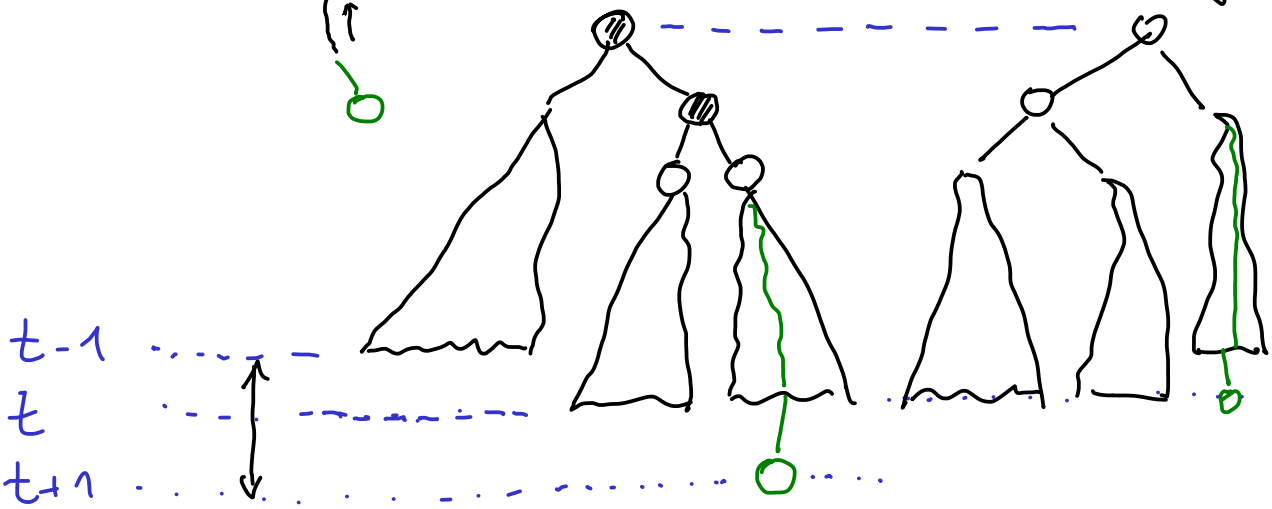
(oder gespiegelt)



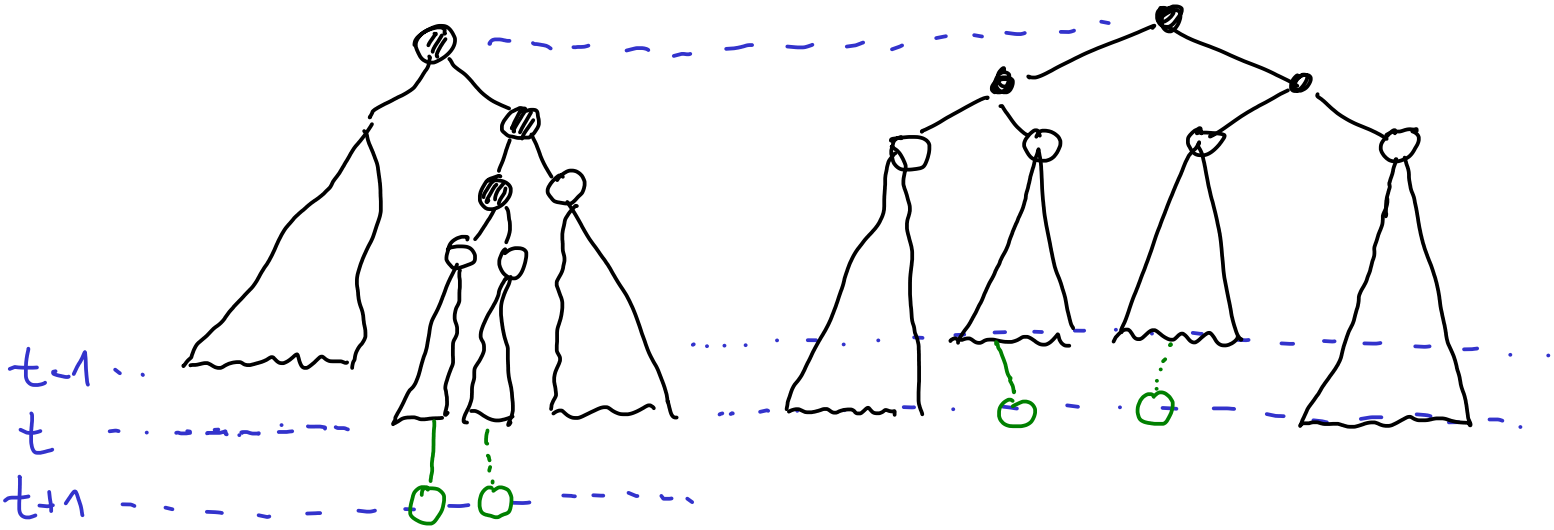
Einfügen

erste Stelle im Pfad vom neu eingefügten Knoten zur Wurzel wo die AVL-Eigenschaft verletzt ist.

Fall 1: Tiefe ist im äußeren Teilbaum gewachsen.



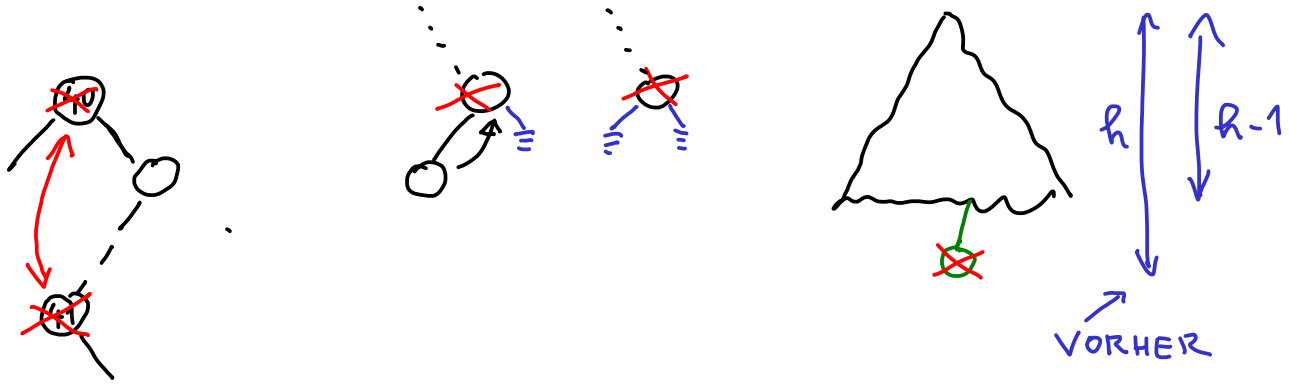
Fall 2: Tiefe ist im mittleren Teilbaum gewachsen.

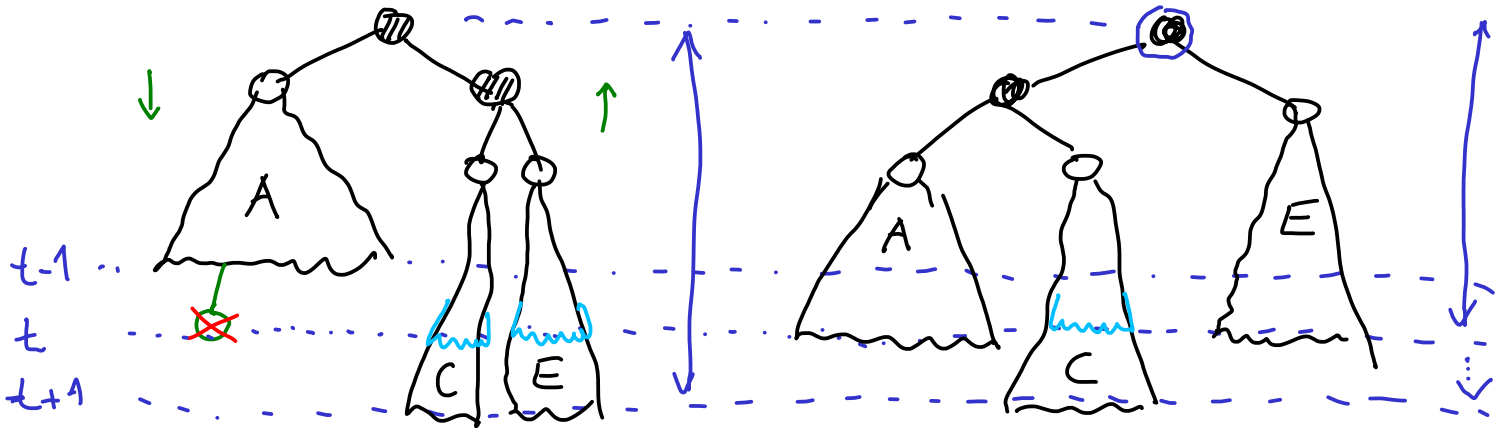


Nach maximal einer Rotation oder Doppelrotation ist die AVL-Eigenschaft wiederhergestellt.

Laufzeit $O(\log n)$

Löschen in AVL-Bäumen

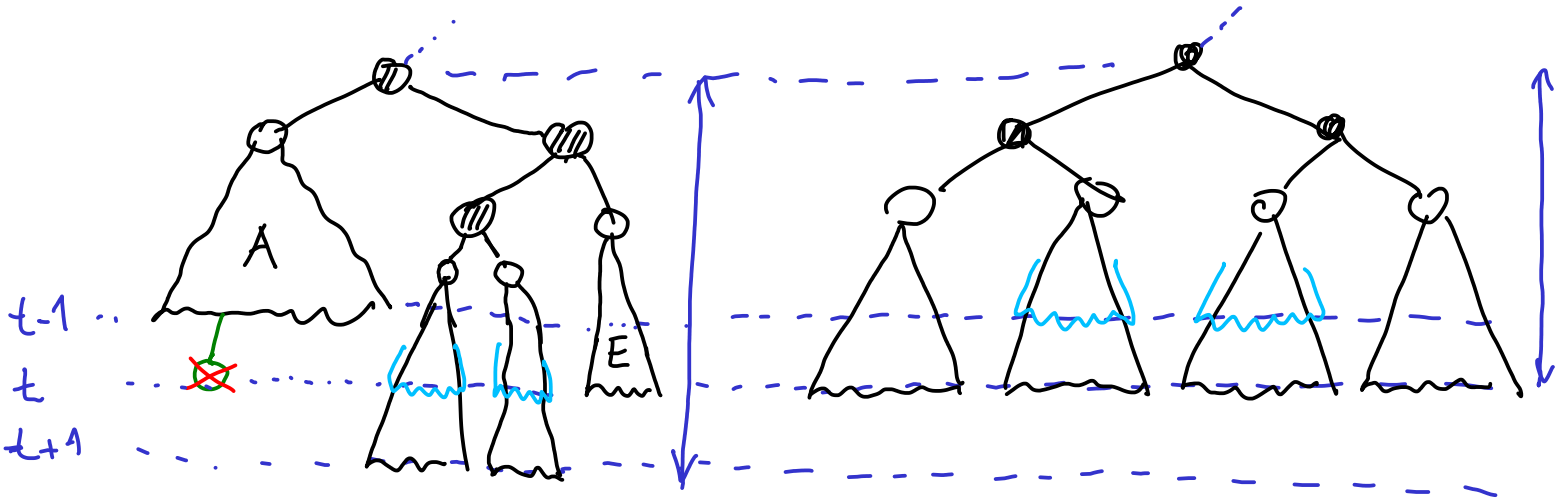




Fall 1: E reicht bis zur Tiefe $t+1$.

Wir müssen eventuell den Pfad bis zur Wurzel weiter gehen und die Knoten überprüfen. (Falls die Tiefe des betrachteten Teilbaums um 1 gesunken ist.)

Fall 2: E reicht bis zur Tiefe t



SATZ

Suchen, Einfügen, Entfernen in einem AVL-Baum mit n Schlüsseln geht in

$$O(\log n)$$

Zeit.

Beim Einfügen muss ma höchstens eine Rotation oder Doppelrotation ausführen.

Beim Entfernen muss ma bis zu $O(\log n)$ Rotationen oder Doppelrotationen ausführen.

Jeder Knoten speichert die Höhe seines Teilbaumes.

(Es reicht auch, die Differenz der Höhen der beiden Teilbäume: $-1, 0,$ oder $+1$ .... 2 Bits)

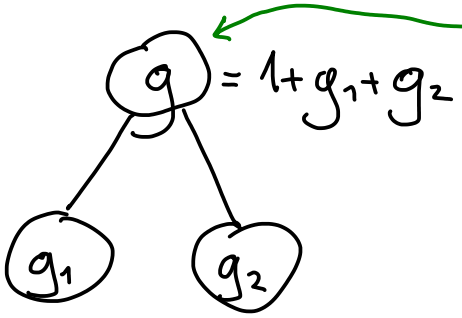
Erweiterte Abfragen in Suchbäumen

Beispiel: Rang: Wie viele Schlüssel sind $< x$?

(Ist x das kleinste, 2-t-kleinste, 3-t-kleinste, ...)

inverse Rangabfrage: Bestimme das k-kleinste.

Jeder Knoten speichert zusätzlich die Größe g des zugehörigen Teilbaums.



suche k-kleinstes in diesem Teilbaum

- $k \leq g_1 \Rightarrow$ linker Teilbaum
- $k = g_1 + 1 \Rightarrow$ Knoten selbst
- $k > g_1 + 1 \Rightarrow$ suche das $(k - g_1 - 1)$-kleinste im rechten Teilbaum

- g kann nach einer Rotation in konstanter Zeit neu berechnet werden.