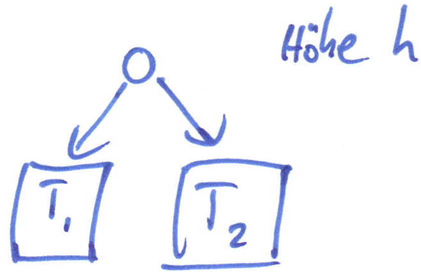
Jetzt noch einmal:



Einer der Teilbäume hat Höhe $h-1$, der andere mindestens Höhe $h-2$.

Daraus ergibt sich

$$n(h) = 1 + n(h-1) + n(h-2).$$

Jetzt zeigen wir per Induktion:

$$\boxed{\text{Behauptung:} \quad n(h) \geq 2^{\frac{h}{2}-1}}$$

Beweis:

Induktionsanfang: Die Behauptung gilt für

$h=1$: $n(1) = 1 \geq 2^{-\frac{1}{2}} = 2^{\frac{h}{2}-1}$

$h=2$: $n(2) = 2 \geq 2^{0} = 2^{\frac{h}{2}-1}$

Induktionsschritt: Gelte die Behauptung für alle $h \in \{1, \dots, k\}$.

Wir zeigen: Die Behauptung gilt für $k+1$!

Denn: $$n(k+1) = 1 + n(k) + n(k-1)$$

Da $n(h)$ monoton wächst (für größere Höhe braucht man mindestens genausoviele Knoten!), gilt

$$n(k) \geq n(k-1)$$

Also
$$\begin{aligned} n(k+1) &\geq 1 + 2 \cdot n(k-1) \\ &\geq 1 + 2 \cdot 2^{\frac{k-1}{2}-1} \\ &= 1 + 2^{\left(\frac{k-1}{2}+1-1\right)} \\ &= 1 + 2^{\left(\frac{k+1}{2}-1\right)} \\ &> 2^{\frac{k+1}{2}-1} . \end{aligned}$$

Damit gilt die Behauptung auch für $k+1$.

Induktionsschluss: Die Behauptung gilt für alle $h \in \mathbb{N}$.

Also $\quad n(h) \geq 2^{\frac{h}{2}-1}$

und damit $\quad \log(n(h)) \geq \frac{h}{2} - 1$

oder $\quad h \leq 2 \log n + 2$.

Damit hat ein AVL-Baum ~~der Höhe~~ mit n Knoten höchstens Höhe $(2 \log n + 2)$, also $O(\log n)$.

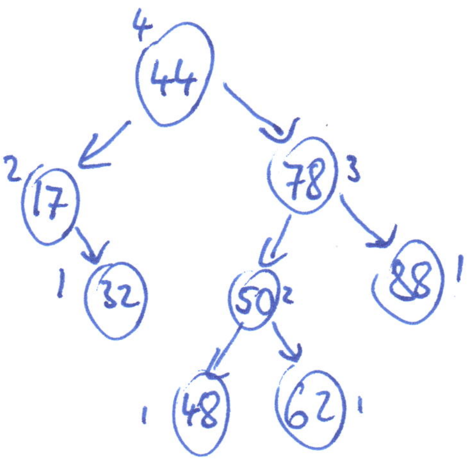
$\square$

Jetzt zurück zu Problem:

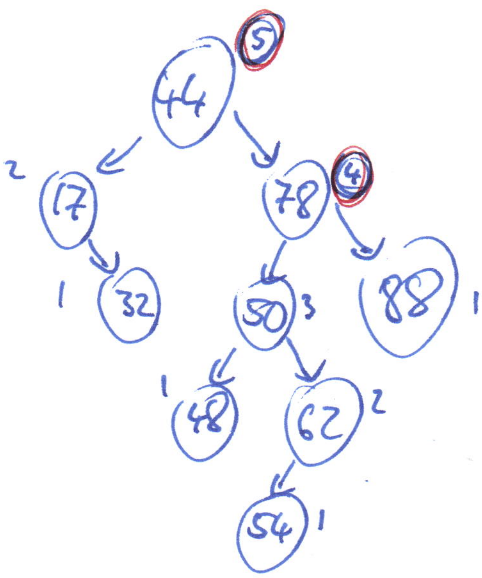
Wie erhält man bei Löschen und Einfügen die Höhenbalanciertheit?

Idee 5: Beim Einfügen oder Löschen ändert sich die Eigenschaft nur lokal und nur ein bisschen → nimm begrenzte lokale Änderungen zur Reparatur vor!

Beispiel:



Füge 54 ein!

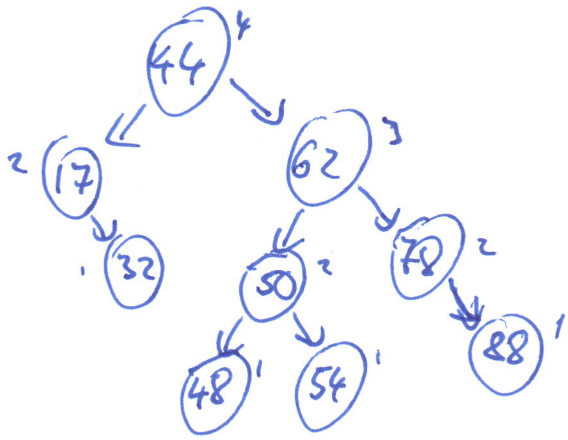


Was ist zu tun?

- Teilbaum der 78 ist nicht höhenbalanciert
- Höhe sollte höchstens 3 sein, damit auch der ganze Baum unter 44 höhenbalanciert ist

→ ~~Verdeutl.~~ Betrachte 78, Kind 50, Enkelkind 62 (Kritischer Pfad unter 78)

Neuer Baum:



- höhenbalanciert!
- nur lokale Umsortierung von 78, 62, 50

Vorher: 78 oben, darunter 50, 62

Jetzt: 62 oben, darunter 78, 50

↳ Rotation!

Mehr Details und Fallbetrachtung folgen...

Genauer: Betrachte Einfügen eines Knotens v:

(Im Beispiel: 54)

Wenn Baum weiter höhenbalanciert → Ok!

Wenn Baum nicht mehr höhenbalanciert

→ Vorfahre von v hat Gewicht dazubekommen, was zu Unbalanciertheit geführt hat

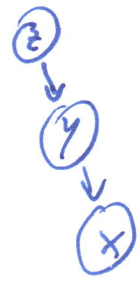
(Im Beispiel sind das 44 und 78)

Sei z der niedrigste unbalancierte Vorfahre von v (Im Beispiel: 78) → Mindestens Höhe 3

Sei y das Kind von z, das Vorfahre von v ist (im Beispiel: 50); dann muss y zwei höher sein als das andere Kind von z.

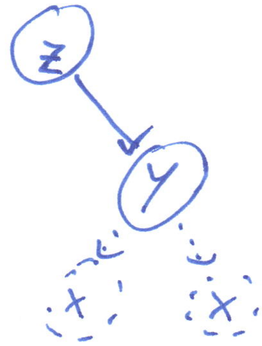
Sei x das Kind von y, das im Teilbaum von v liegt. (im Beispiel: 62)

Also: Großvater z
Vater y
Kind x
(eventuell auch x = v möglich)



Wie können x, y, z zueinander stehen?

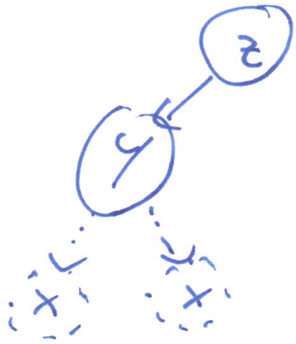
Beobachtung: Wenn $z < y$,



dann auch $z < x$

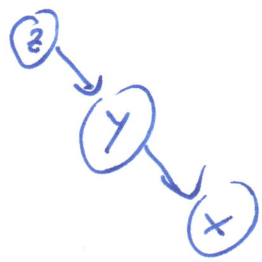
(Suchbaumeigenschaft!)

Wenn $z > y$,
dann auch $z > x$

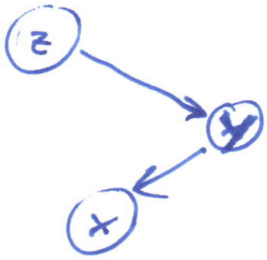


Es bleiben die Möglichkeiten:

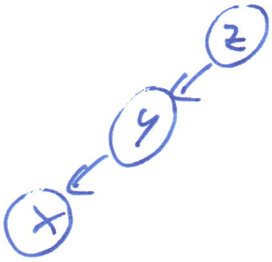
(I) $z < y < x$



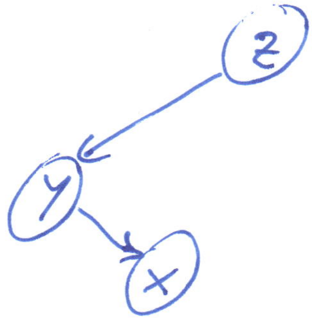
(II) $z < x < y$



(III) $x < y < z$

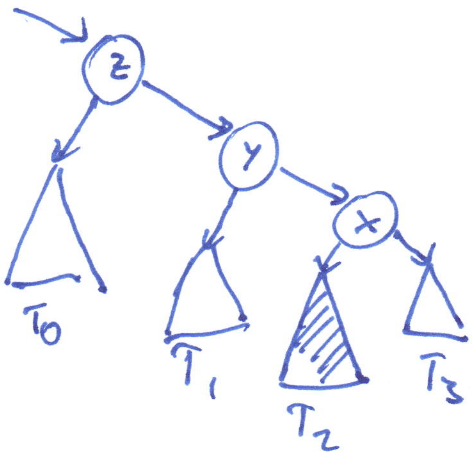


(IV) $y < x < z$

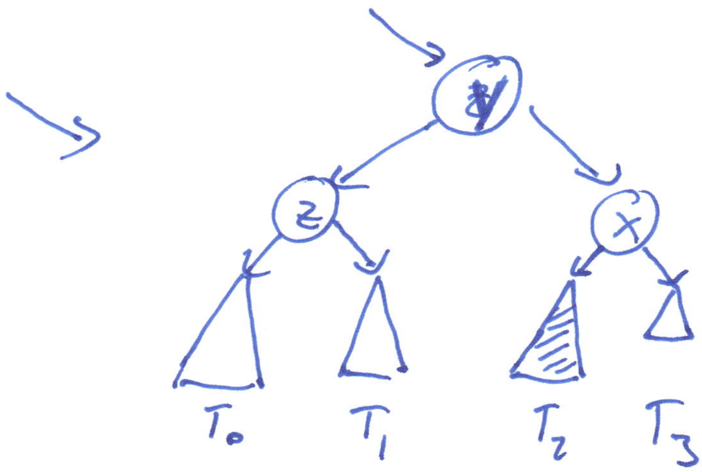


Jetzt betrachten wir folgende Reparaturoperationen:

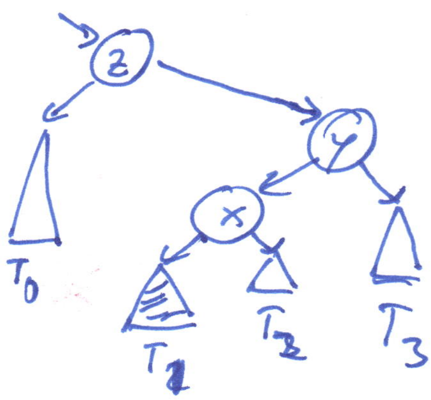
(I)

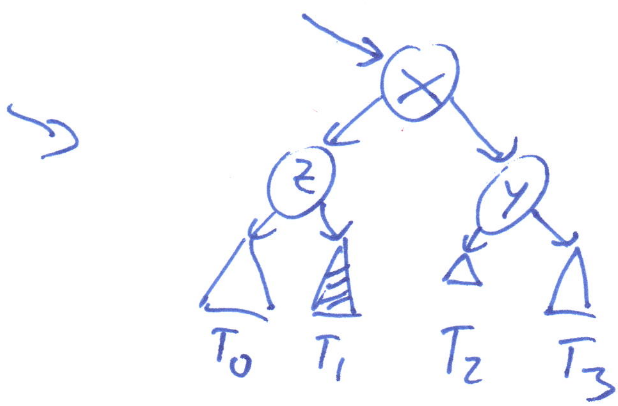


T_2 enthält v, geht zwei Level tiefer als T_0, ein Level tiefer als T_1, T_3



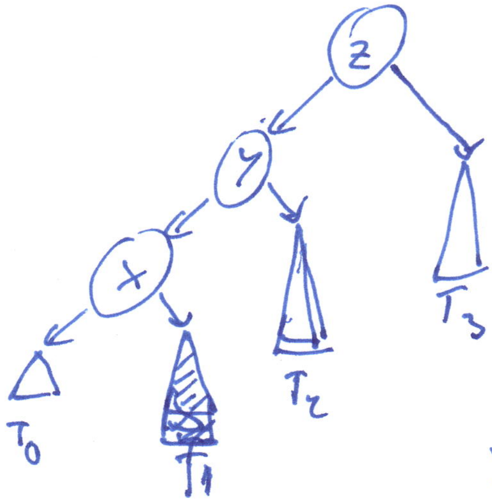
(II)



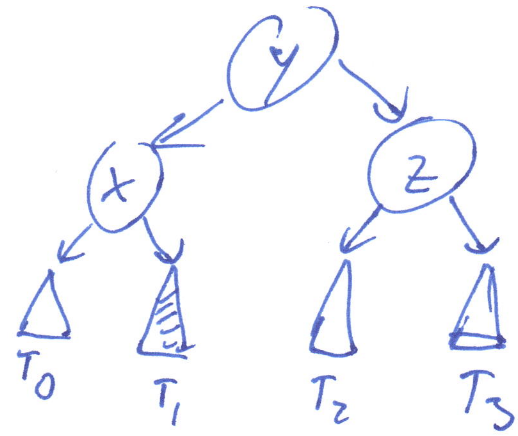


(T_1 zwei Level tiefer als T_0!)

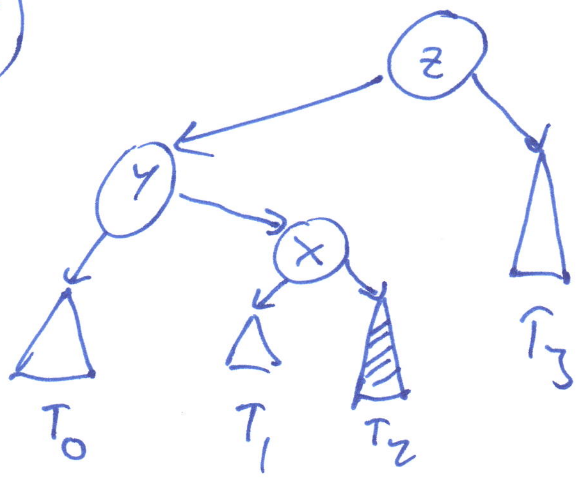
(III)



(T_1 zwei Level tiefer als T_3, eins tiefer als T_0, T_2



(IV)



(T_2 zwei Level tiefer als T_3, eins tiefer als T_0, T_1)