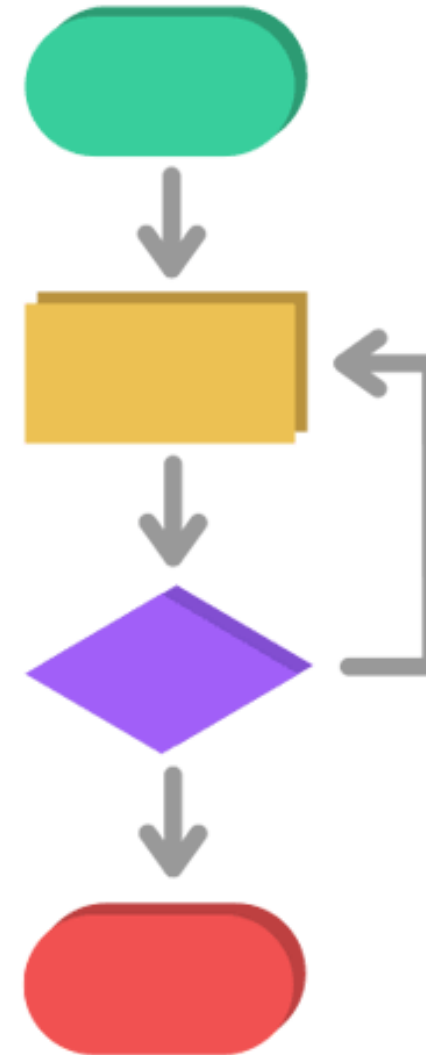


# Algorithmen Tutorium 12

Beginn: 16:15



# Organisation

---

- Letztes ÜB sehr gut verlaufen!
- Ergebnisse des Wettbewerbs kommen noch

# Inhalt

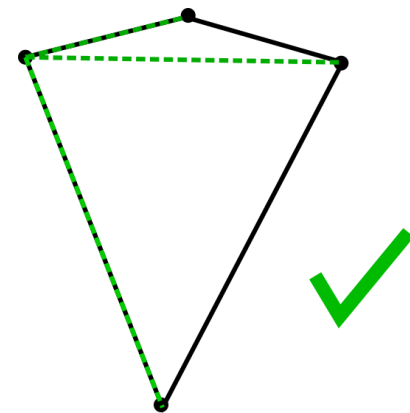
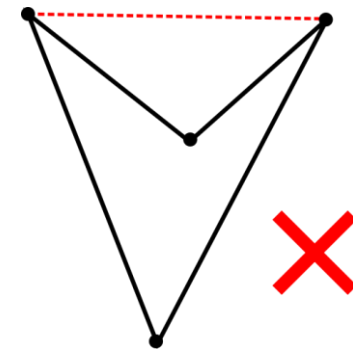
---

- Vllt. Konvexe Hülle
- Balance und Rotieren
- (2,4)-Baum

# Konvexe Hülle

---

- Was heißt konvex(-es Polygon)?
  - Jeder Punkt ist von jedem anderen Punkt im Polygon sichtbar
- Was heißt sichtbar?
  - Direkte Verbindung zweier Punkte schneidet Polygonkante NICHT
  - Okay: Verbindung liegt AUF der Kante
  - Nicht Okay: Verbindung verlässt Polygon
- Polygon aus (minimaler) Teilmenge von Punkten,  
die alle anderen geometrisch konvex einfasst
- Algorithmus: Übung



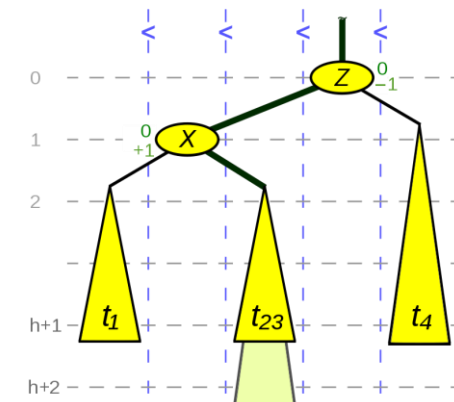
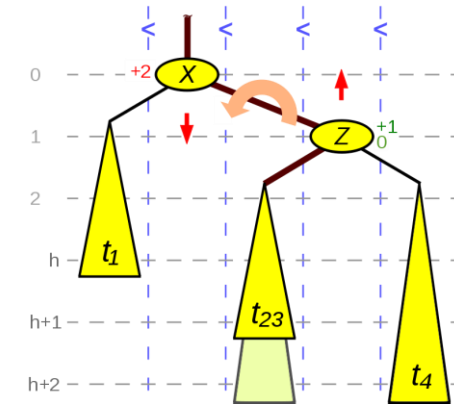
# AVL Bäume (AVL Bedingung)

---

- Für alle Knoten  $v$  (beginnend mit den Blättern)
  - Berechne die Anzahl der Knoten im Teilbaum mit  $v$  als Wurzel
    - 1 Für Blätter
    - Summe der Werte der Kinder + 1
  - Berechne Balance von  $v$
  - Wenn Balance ungültig, stelle AVL Bedingung her (Rebalancierung)
- Operationen zum Rebalancieren
  - Einfachrotation
  - Zweifachrotation
- **Achtung: Rebalancieren ist symmetrisch! Wenn Z links von X ist, ändert sich die Richtung**

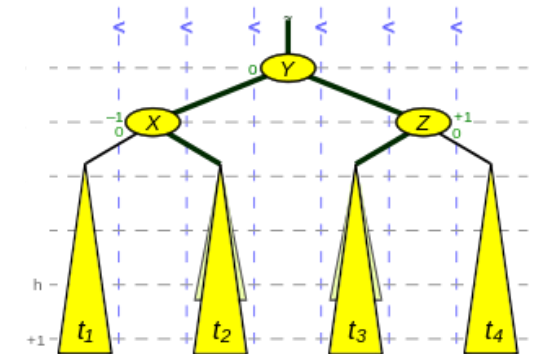
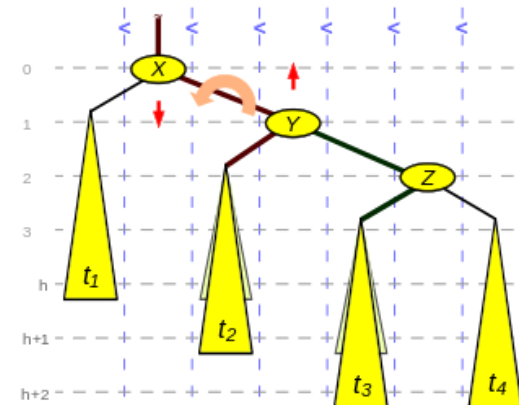
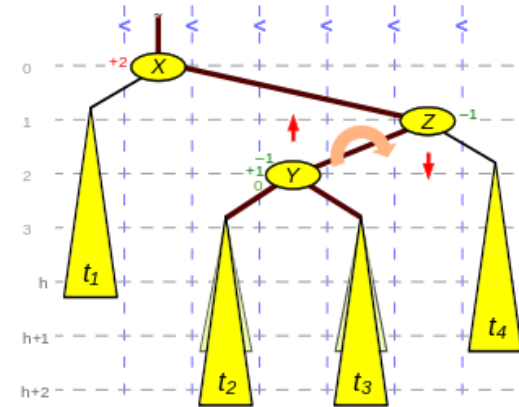
# AVL Bäume (Einfachrotation)

- Problem: Knoten X verletzt AVL Bedingung (+2)
  - Der Teilbaum von X mit Wurzel Z ist um 2 höher
  - Der linke Teilbaum von Z **ist nicht höher** als der rechte
- Operation:
  - Teilbaum in Richtung von Z wird „ausgehängt“
  - Tausche Knoten X mit Elternknoten Z
  - Hänge den Teilbaum, an den Ast von X an dem vorher Z war, an



# AVL Bäume (Zweifachrotation)

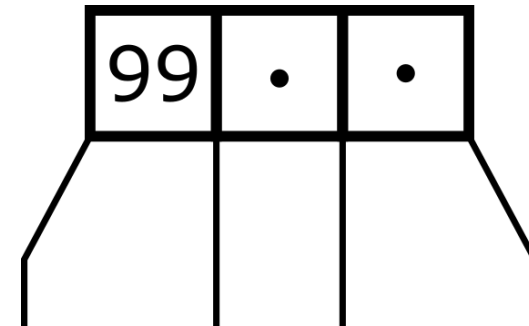
- Problem: Knoten X verletzt AVL Bedingung (+2)
  - Der Teilbaum von X mit Wurzel Z ist um 2 höher
  - Der linke Teilbaum von Z hat Y als Wurzel
  - Der linke Teilbaum von Z **ist höher** als der rechte
- Operation:
  - Einfachrotation von Y und Z
  - Einfachrotation von Y und X



# (2,4)-Baum

---

- Baum mit mind. 2 und max. 4 Kindern pro Knoten
  - Wurzel mind. 2 Kinder
- Alle Blätter mit gleicher Tiefe
- 3 Vergleichsschlüssel(Keys) pro Knoten
- Keys im Knoten geordnet
- Knoten werden üblicherweise linksbündig dargestellt
- Bei weniger Kindern können Blätter/Schlüssel auch **null/nil** sein





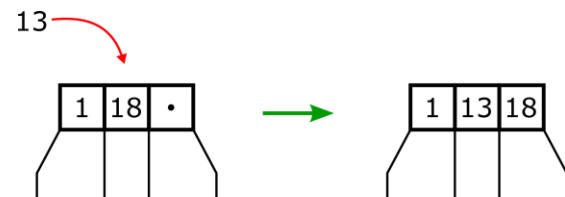
# (2,4)-Baum

---

- Werte werden in den Blätter gespeichert
  - NICHT in den Knoten (nur Vergleichsschlüssel)
- Schlüssel repräsentieren größtes Blatt im jeweiligen Unterbaum (NICHT beliebig)
  - Größter Wert wird als Schlüssel in Wurzel gespeichert
- Wird beim Einfügen & Löschen Balanciert
  - Suchen in konstant  $O(\log(n))$

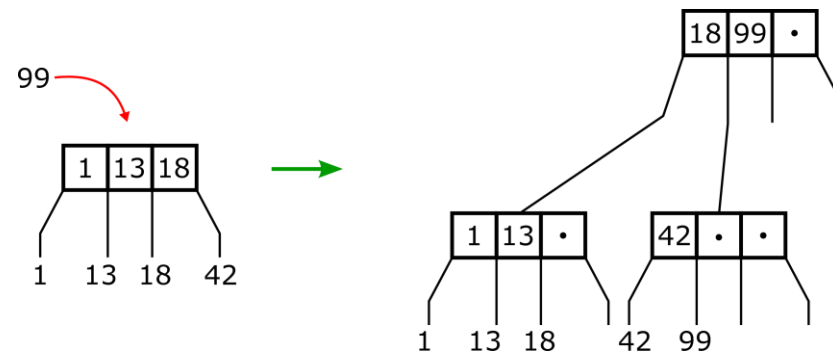
# (2,4)-Baum – Einfügen

- Triviales Einfügen



- Einfügen, bei „vollem“ Knoten

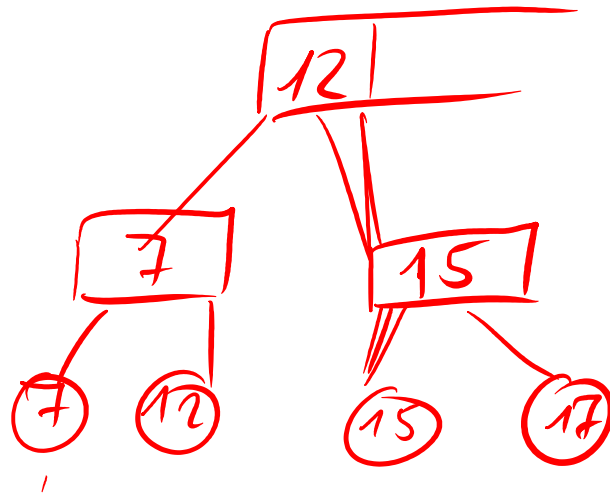
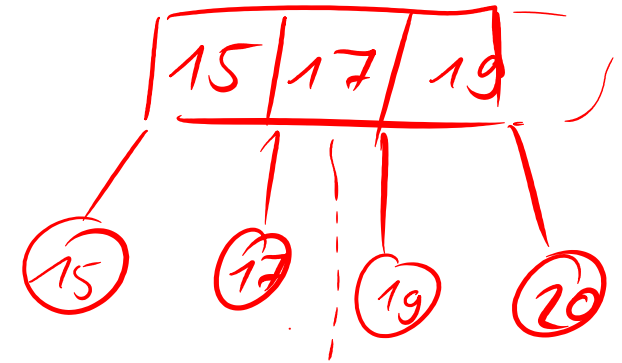
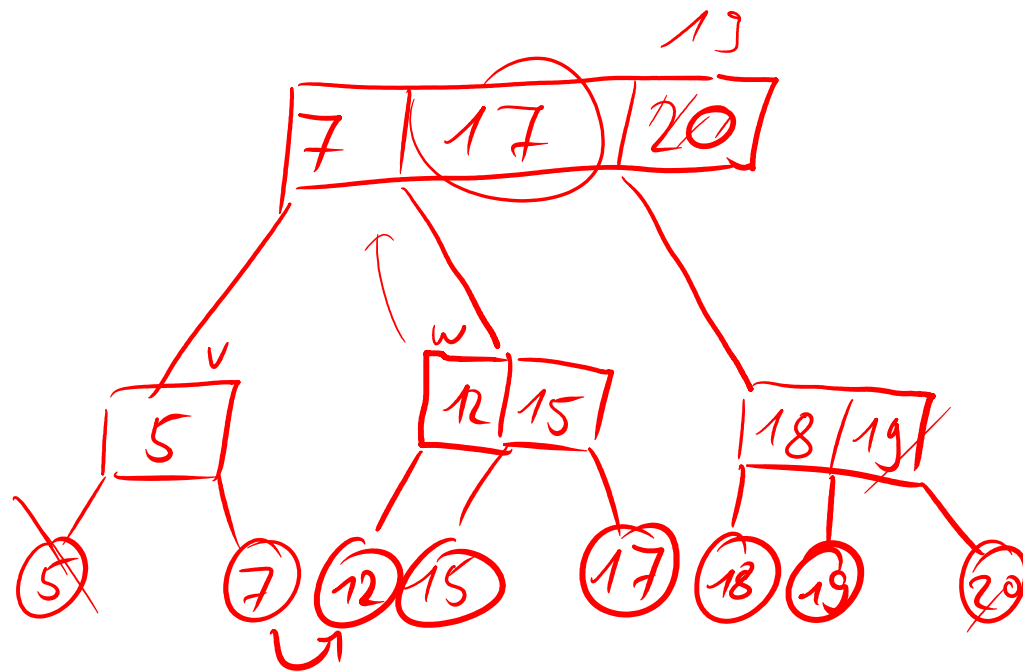
- Wichtig: Wenn nicht die Wurzel, wird 18 beim Elternknoten eingefügt



# (2,4)-Baum – Löschen

---

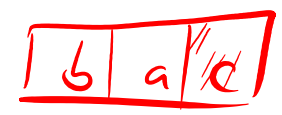
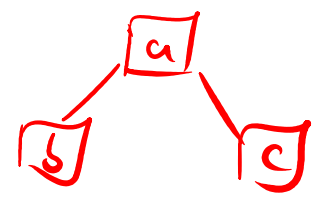
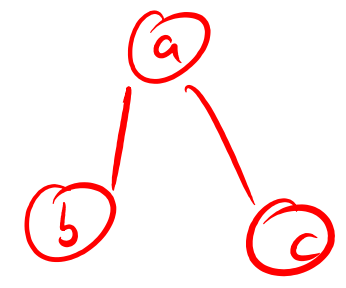
- Weitestgehend analog zum Einfügen (Schlüssel ggf. Löschen)
- Sonderfall: Knoten  $v$  hat nur ein Kind
  - Geschwister  $w$  hat genau 2 Kinder
    - $\Rightarrow$  Verschmelze  $v$  und  $w$  & Lösche Schlüssel rekursiv bei  $\text{parent}(v)$
  - Geschwister  $w$  hat  $\geq 3$  Kinder
    - $\Rightarrow$  Stehle Nächsten Knoten von  $w$  zu  $v$  (Passe Schlüssel an)



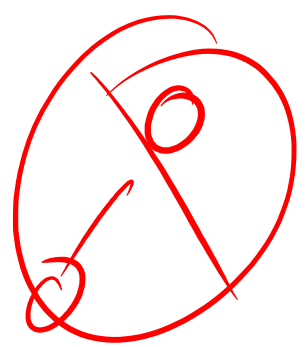
1A:



$$h = 0$$

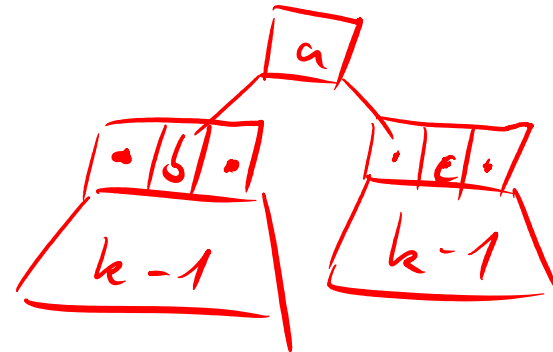
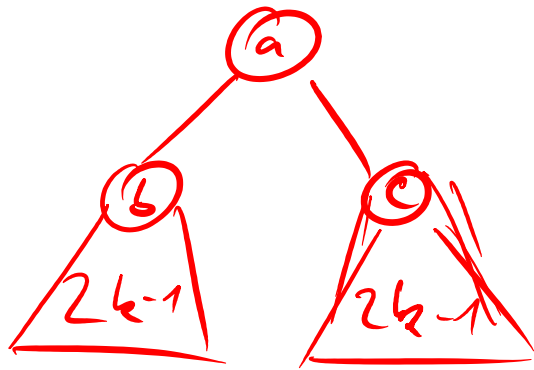


$$h = 1$$

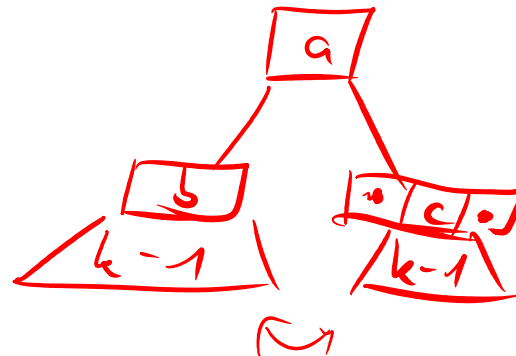
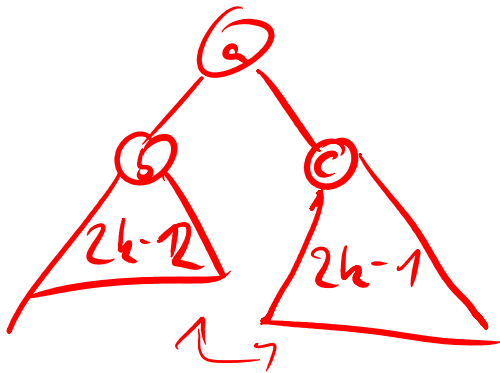


<sup>13</sup> V: Gelte für  $h+1 \quad \forall h \geq 0$ .

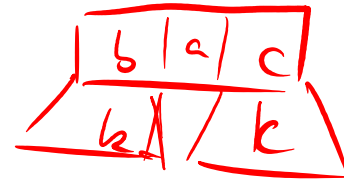
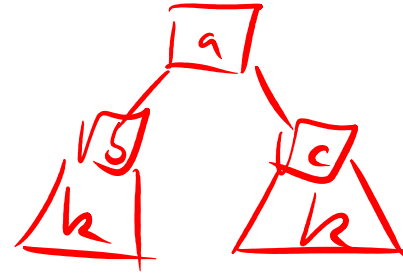
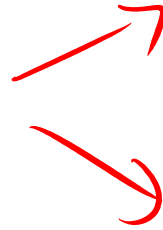
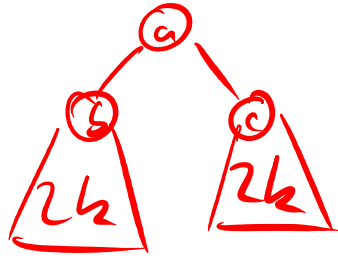
B: I:  $h=2k$  &  $Bal(a)=0$



II:  $h=2k$  &  $|Bal(a)|=1$



III :  $b = 2k + 1$  &  $Bal(a) = 0$  ;



IV :  $b = 2k + 1$  &  $|Bal(a)| = 1$

