a-b-Bäume, 2-3-Bäume

$a \geq 2$ $\quad b \geq 2a-1$

Beispiele:

$a=2,\ b=3$

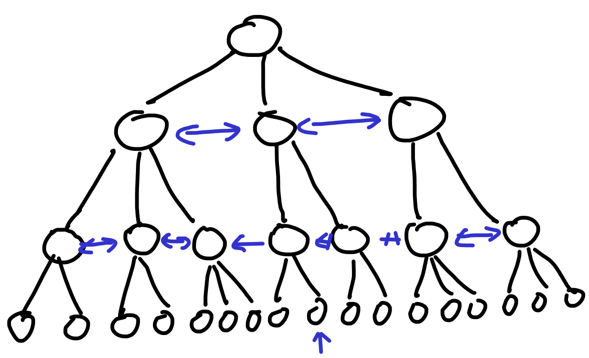
$2, 4$

$2, 7$

$1000, 1999$

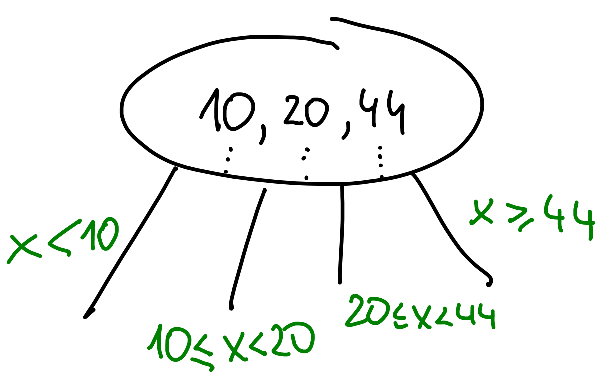
Suchbäume:

- Alle Blätter haben die gleiche Tiefe.
- Alle inneren Knoten haben $\geq a$ und $\leq b$ Kinder.
  Ausnahme: die Wurzel hat $\geq 2$ und $\leq b$ Kinder.



- Alle Werte sind nur in den Blättern gespeichert; die inneren Knoten enthalten nur Schlüssel.

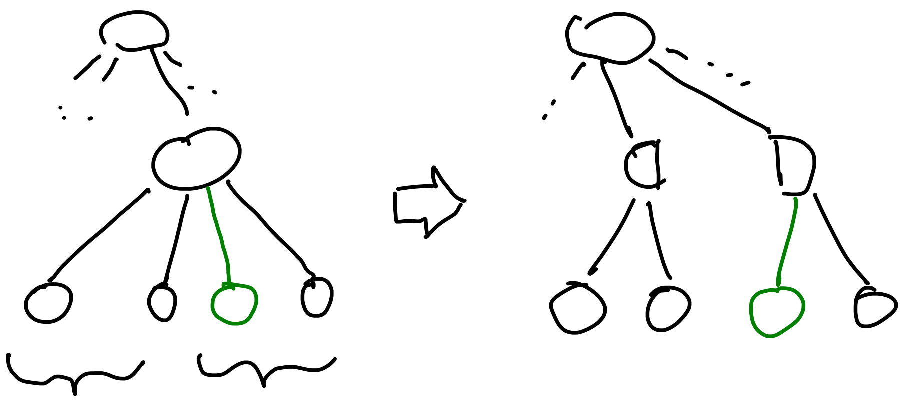
(Es gibt auch Varianten.)




Ein Knoten mit k Kindern enthält k-1 Schlüssel.

SUCHEN: $O(\log n)$ ✓

EINFÜGEN: ... passende Stelle suchen und Blatt einfügen.



Ein Knoten kann (zeitweilig) $\underline{> b}$ Kinder haben.

$b+1 \geq 2a \Rightarrow$ Aufspalten in 2 Knoten mit $\geq a$ Kindern.

Der Elternknoten kann dadurch überlaufen ... $\Rightarrow$ Aufspalten

... usw. bis zur Wurzel.

Wurzel aufspalten $\Rightarrow$ Höhe des Baumes steigt um 1.

Laufzeit $O(\log n)$

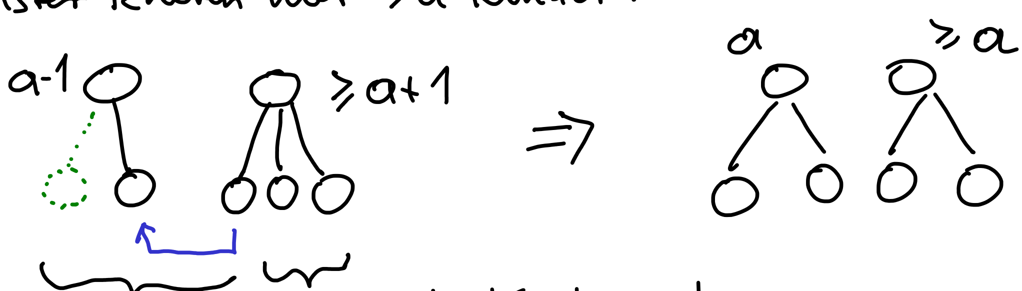
---

LÖSCHEN ... Blatt suchen und entfernen

Ein Knoten kann (zeitweilig) $< a$ Kinder haben.

$a-1$ Kinder.

- Ein Geschwisterknoten hat $> a$ Kinder.




ein Kind ausborgen.

- Ein Geschwisterknoten hat $a$ Kinder.

$$(a-1) + a = 2a-1 \leq b \checkmark$$

Elternknoten verliert ein Kind ... usw. Richtung Wurzel

Wurzel hat nur 1 Kind $\Rightarrow$ Höhe des Baumes sinkt um 1.

$$\boxed{\text{Alle Wörterbuchoperationen können in } O(\log n) \text{ Zeit durchgeführt werden.}}$$

---

B-Bäume : a groß $B^*$, $B^+$,

- <u>große</u> Knoten : 1 Knoten = 1 Block im externen Speicher

Hauptkriterium : Anzahl der Zugriffe auf Knoten (Latenzzeit!)

Höhe $\leq \log_a n$