

Definition

- Art von a-b-Baum
 - Spezialfall von [[Balancierte Bäume]]
 - ähnlich wie [[Binary Search and BTree]]
- Eigenschaften
 - leicht zu halten bei blattorientierter Speicherung
 - * anstatt knotenorientiert
 - worst case optimal
 - (1) Alle Äste sind gleich lang.
 - (2) Die max. Anzahl der Kinder eines Knotens ist 4.
 - (3) Innere Knoten haben ≥ 2 Kinder.
 - (4) Die Blätter enthalten v.l.n.r. die Werte aufsteigend sortiert.
 - (5) Jeder innere Knoten k mit $\alpha(k)$ Kinder ($2 \leq \alpha(k) \leq 4$) speichert $\alpha(k)-1$ Hilfsinformationen $x_1, \dots, x_{\alpha(k)-1}$, wobei x_i = größter Wert im Teilbaum des i -ten Kindes von links.

Höhe $h = \Theta(\log n)$

n ...Anzahl der Blätter (= Anzahl gespeicherte Keys)
 h ...Höhe des Baumes

- Anzahl der Blätter wird mit jeder weiteren Schicht
 - Zumindest verdoppelt
 - Maximal vervierfacht
- Bei n Blättern bekommen wir

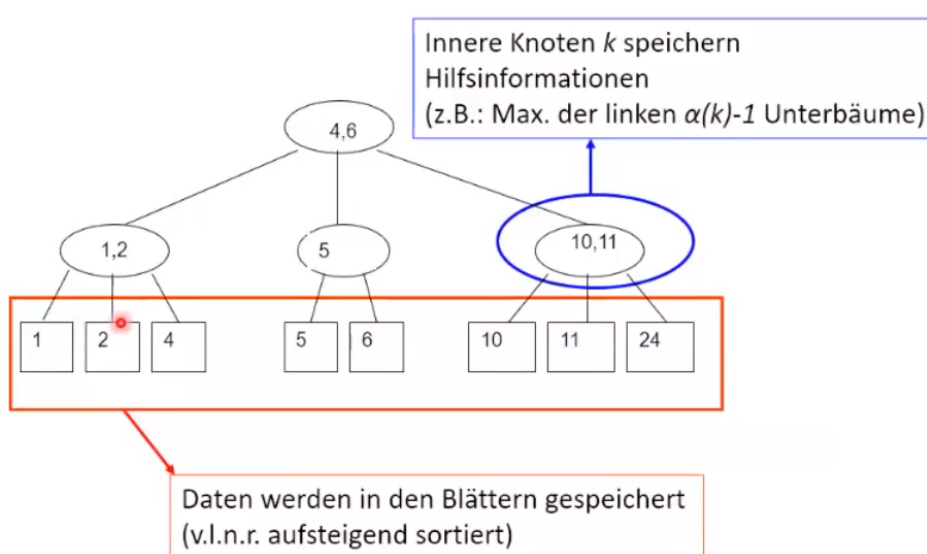
$$2^h \leq n \leq 4^h = 2^{2h}$$

$$h \leq \lg n \leq 2h$$

$$\frac{\lg n}{2} \leq h \leq \lg n$$

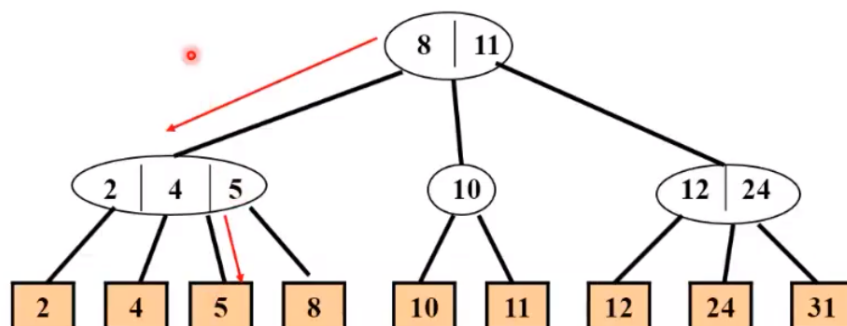
Anzahl der Knoten bei n Blättern:

$$\#Knoten \leq n + \frac{n}{2} + \frac{n}{4} + \dots + 1 < 2n = \Theta(n)$$



Operationen

- **Suchen:** pro Knoten wird der relevante Teilbaum in $O(1)$ Zeit pro Knoten selektiert $\Rightarrow \Theta(h) = \Theta(\log n)$ Zeit

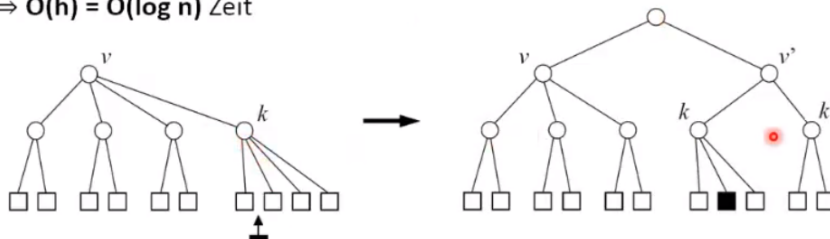


Einfügen: Suchen, Blatt an Knoten k anhängen

- $\alpha(k) \leq 4$: Resultierender Baum ist wieder ein (2-4)-Baum
- $\alpha(k) = 5$: Resultierender Baum ist **kein** (2-4)-Baum

SPALTEN von k : Gib k einen Bruder k' rechts von k . Hänge die 2 rechtesten Kinder von k auf k' um $\Rightarrow \alpha(k) = 3, \alpha(k') = 2$

SPALTEN muss evtl. für übergeordnete Knoten wiederholt werden (evt. bis zur Wurzel. Diese wird dann Kind einer neuen Wurzel mit $\alpha(w)=2$) $\Rightarrow \Theta(h) = \Theta(\log n)$ Zeit



• **Entfernen:** Suchen, Blatt von Knoten k entfernen

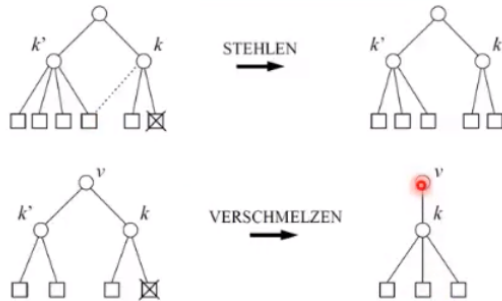
– $\alpha(k) \geq 2$: Resultierender Baum ist wieder ein (2-4)-Baum

– $\alpha(k) = 1$: Resultierender Baum ist **kein** (2-4)-Baum

Sei k' ein direkter Bruder von k :

– $\alpha(k') \geq 3$: **STEHLEN** eines Kindes von $k' \Rightarrow \alpha(k) = 2, \alpha(k') \geq 2$

– $\alpha(k') = 2$: **VERSCHMELZEN** von k mit $k' \Rightarrow \alpha(k) = 3$



Verschmelzen evt. für übergeordnete Knoten wiederholen (Wurzel wird durch einziges Kind ersetzt) $\Rightarrow O(h) = O(\log n)$ Zeit

Zusammenfassung

Der **(2-4)-Baum** ist eine Datenstruktur, die das **Wörterbuchproblem** (Suchen, Einfügen, Entfernen) auf einer Menge von n Elementen in **$O(\log n)$ Zeit** pro Operation löst, und **$O(n)$ Speicher** belegt.

Dies ist worst-case optimal (auch bzgl. statischer Suche!).

Einfügen und Entfernen erfordert **Umstrukturierungen** (Spalten, Stehlen, Verschmelzen)

Bereits vorgenommene Umstrukturierungen **amortisieren**

• sich jedoch später.