

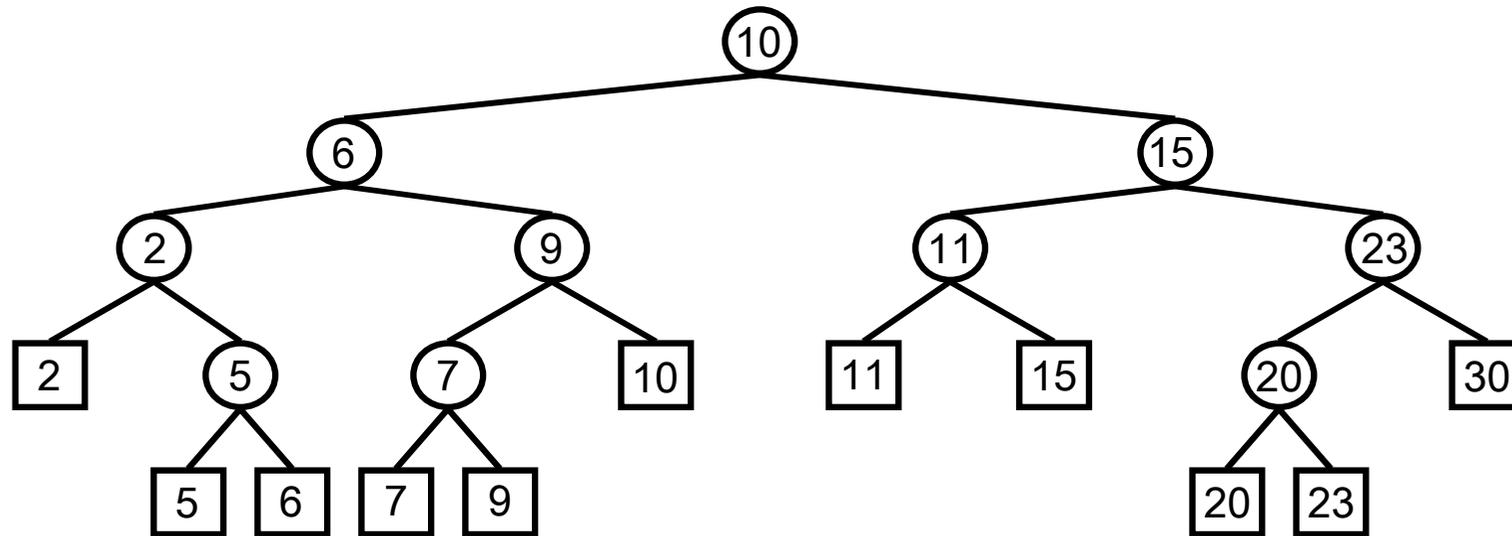
Vorlesung Informatik 2

Algorithmen und Datenstrukturen

(24 – Relaxiertes Balancieren)

Prof. Dr. Susanne Albers

Balancierte (Blatt-)suchbäume



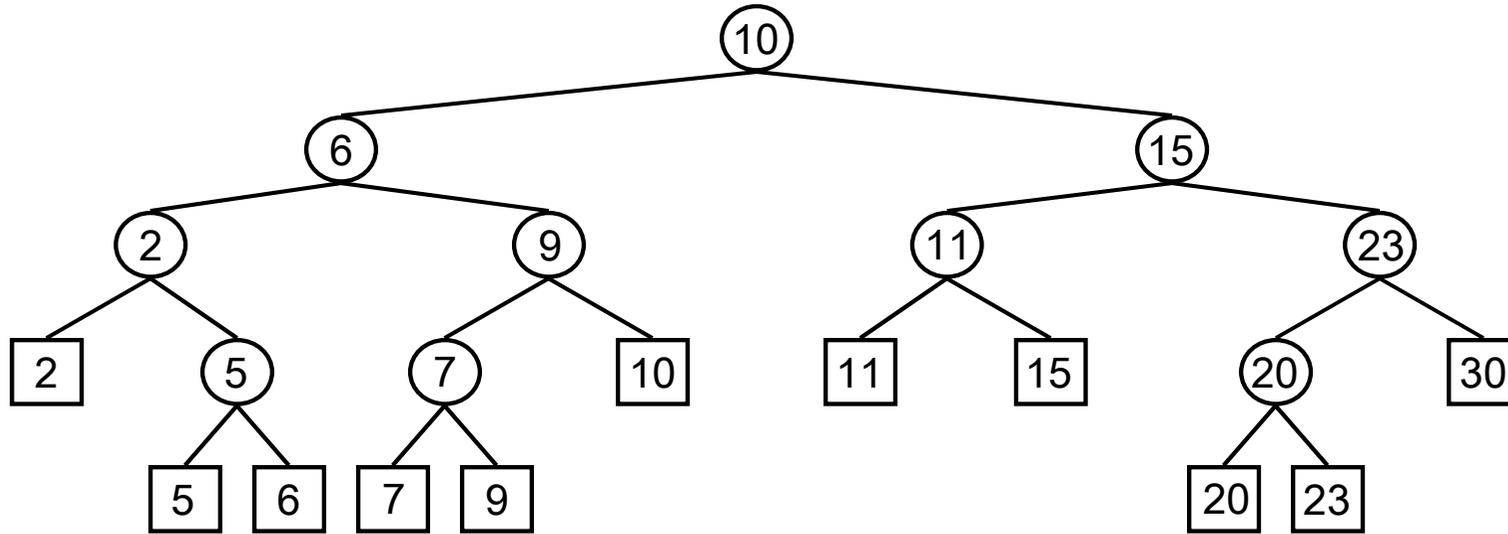
Dictionary: Menge von Schlüsseln,

Operations: Suchen, Einfügen, Entfernen

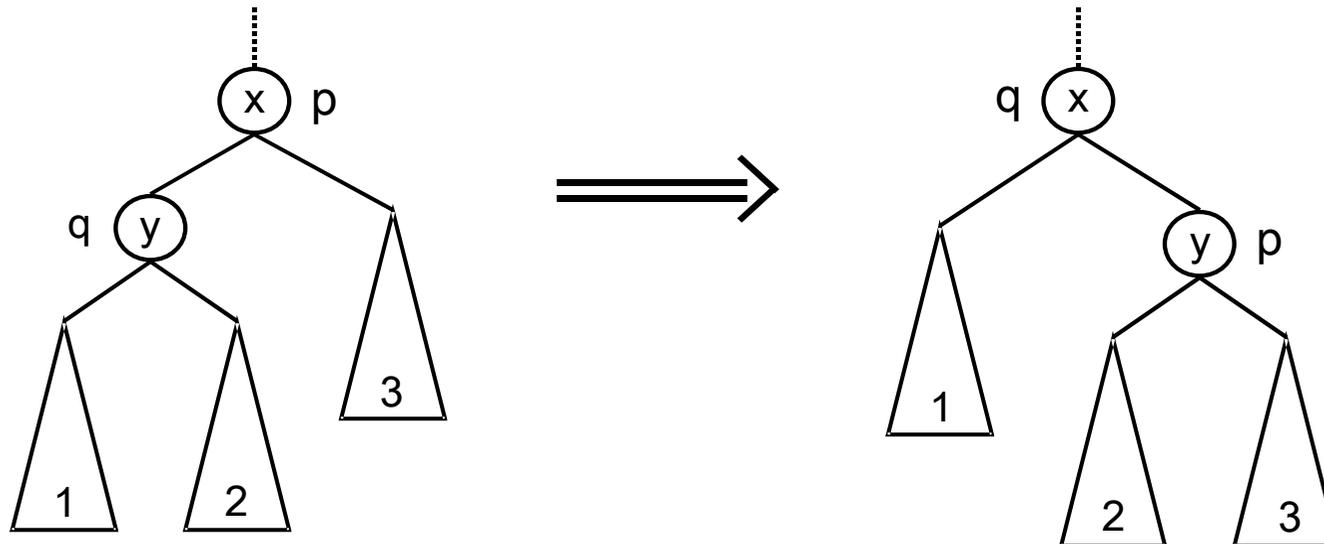
Balancierte (Blatt-)suchbäume sind eine mögliche Implementation von dictionaries.

AVL-trees, symmetric Binary B-Trees, Red-Black-trees, Half-balanced Trees, Weight-balanced Trees, B-Bäume, Bruder-Bäume, ...

Rotation



Rotation



Striktes und relaxiertes Balancieren

Striktes Rebalancieren:

Auf jedes Update folgt unmittelbar eine Folge von Rebalancierungs-Operationen.

Relaxiertes Balancieren:

Entkopplung der Rebalancierungs- von den Update-Operationen, sodass beide nebenläufig ausführbar werden.

Ziel 1: Konstante Umstrukturierungskosten

Ziel 2: Erhöhen der Parallelität von Such- und Update-Operationen

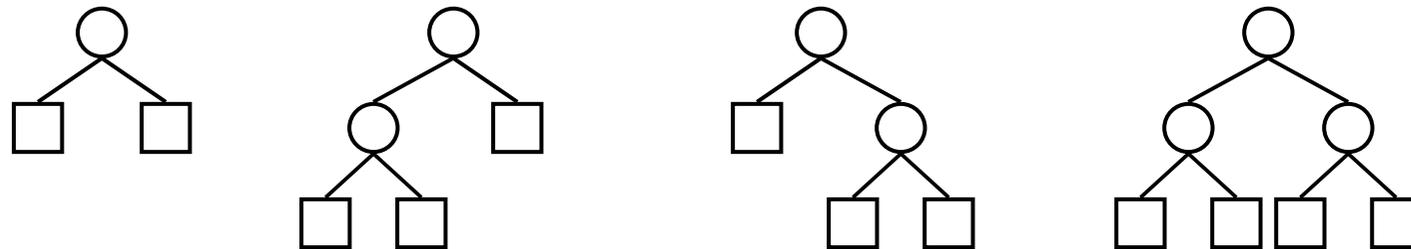
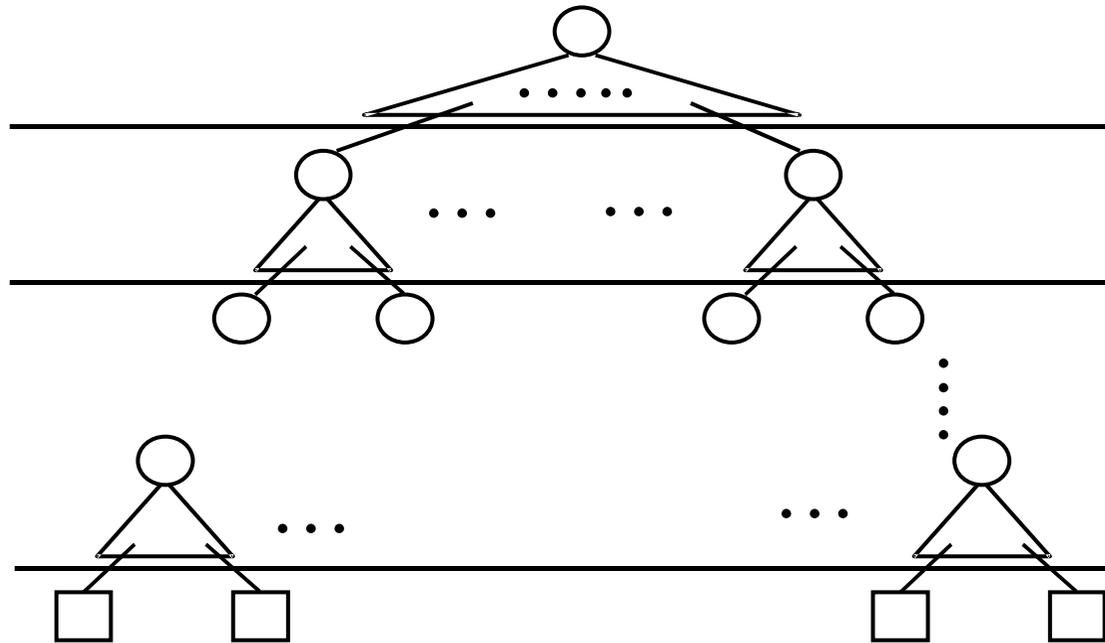
Kessels 1983

Larsen 1994, 1997, ...

Larsen, Fagerberg 1995

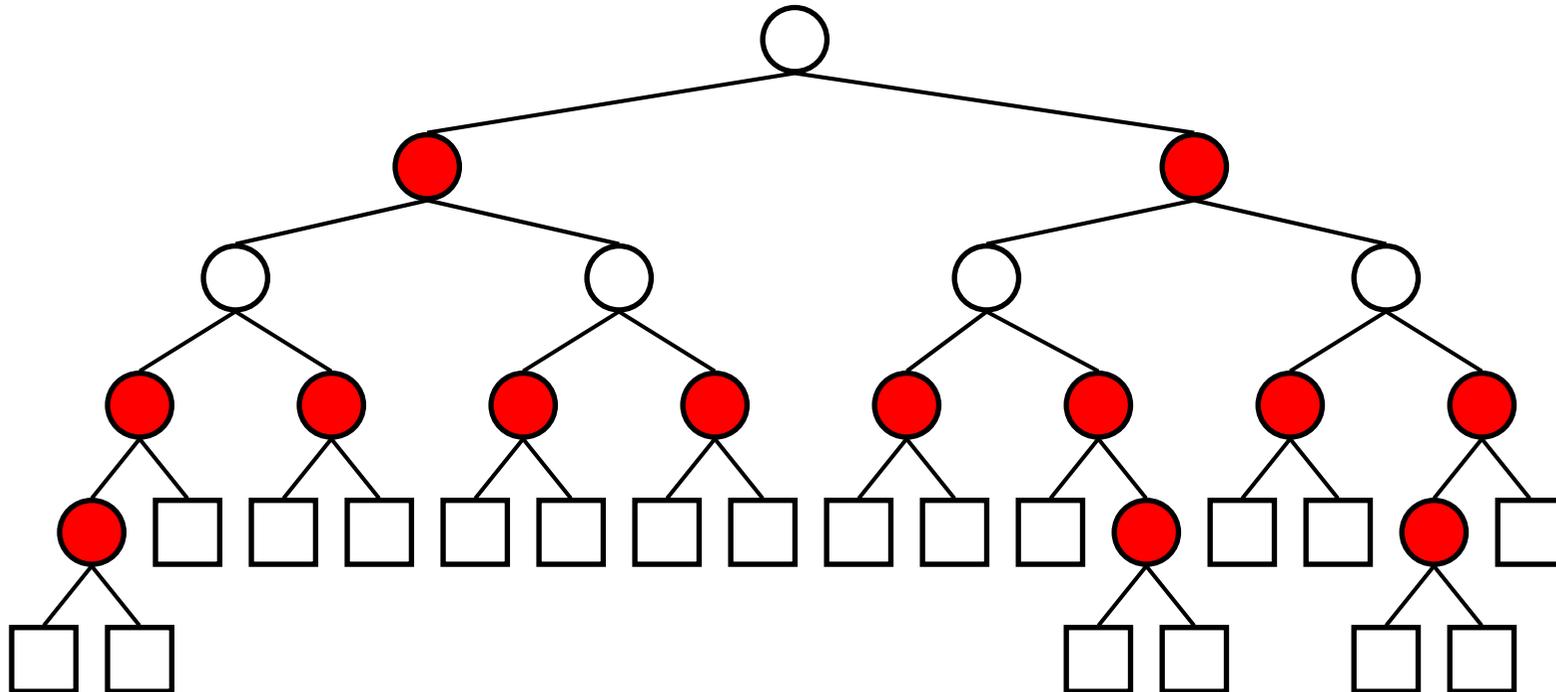
Ot/Soisalon-Soininen/Larsen: Acta Informatica 2001

Stratifizierte Suchbäume

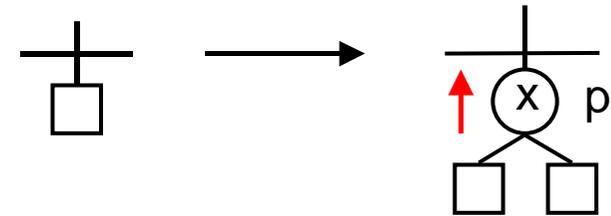
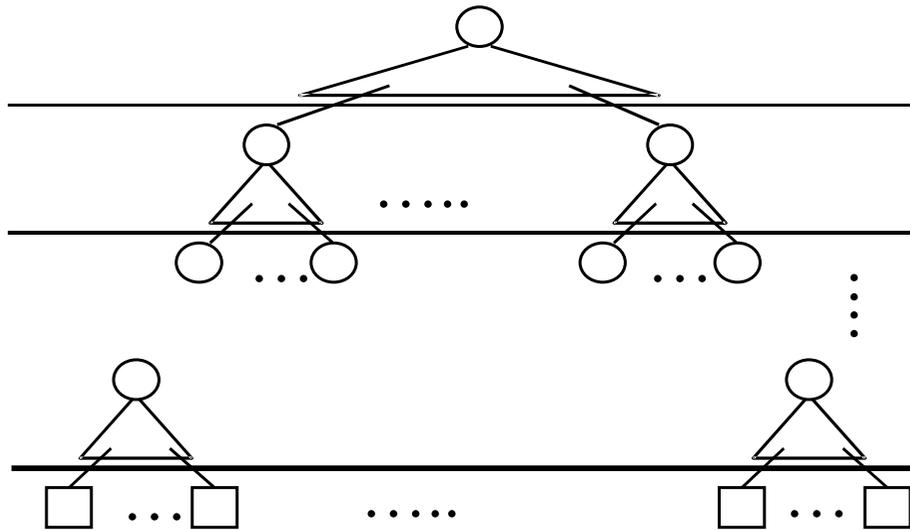


Die Höhe eines Z-stratifizierten Baumes mit N Blättern ist $O(\log N)$

Beispiel

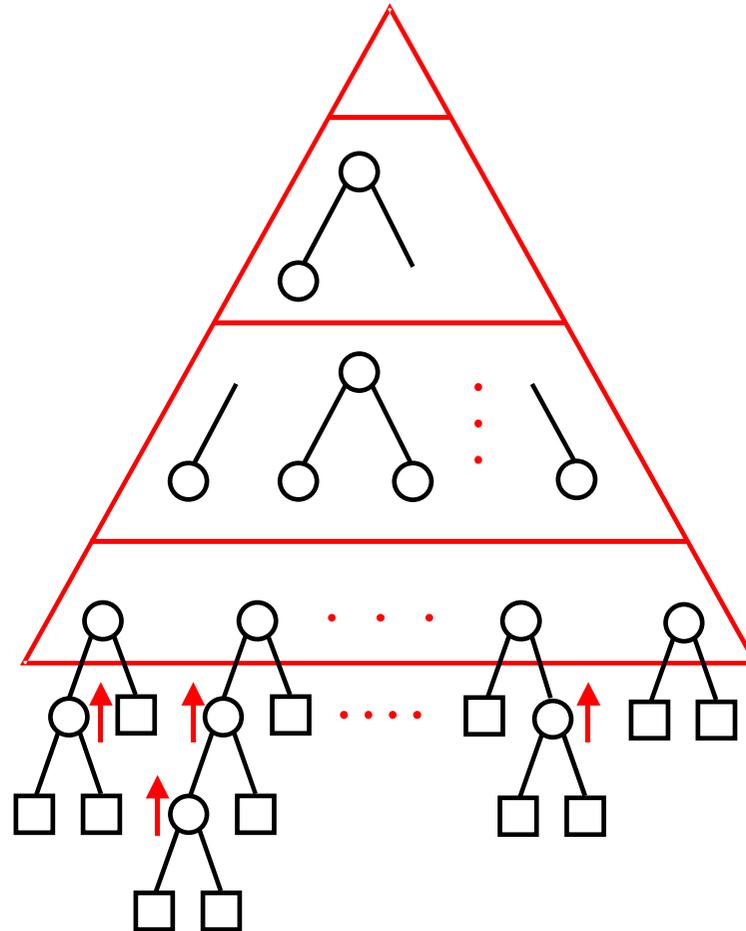


Einfügen



Füge neuen Schlüssel ganz unten ein und deponiere einen „push-up-request“ ↑

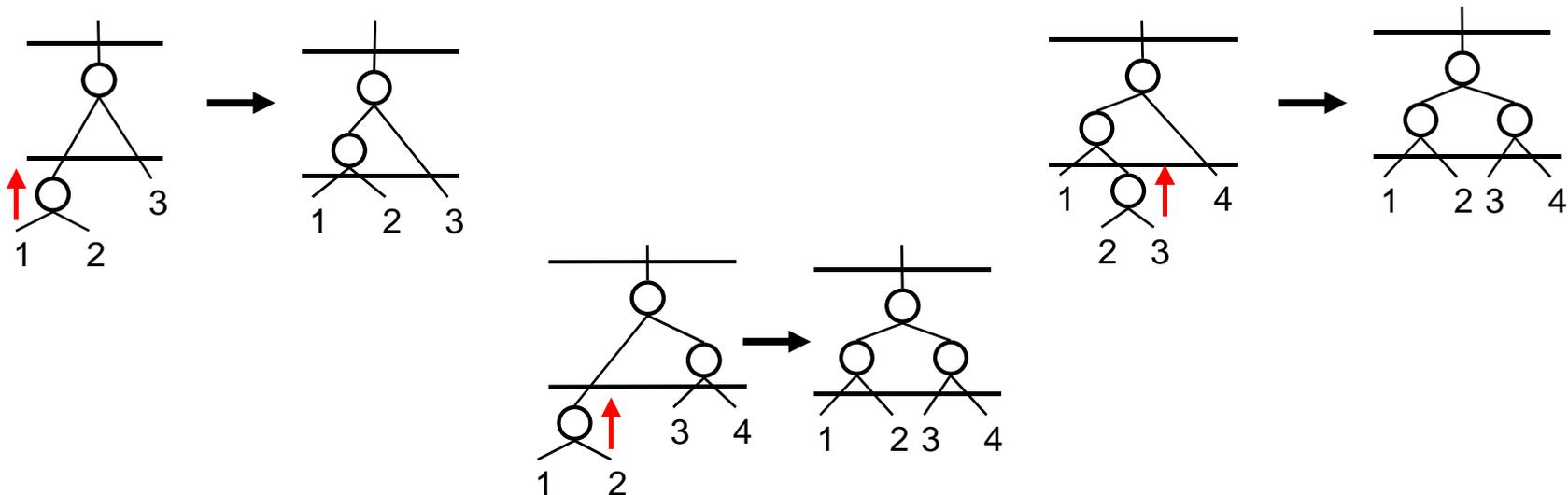
Iterierte Folge von Einfügungen



Beseitigung eines Push-up-requests (1)

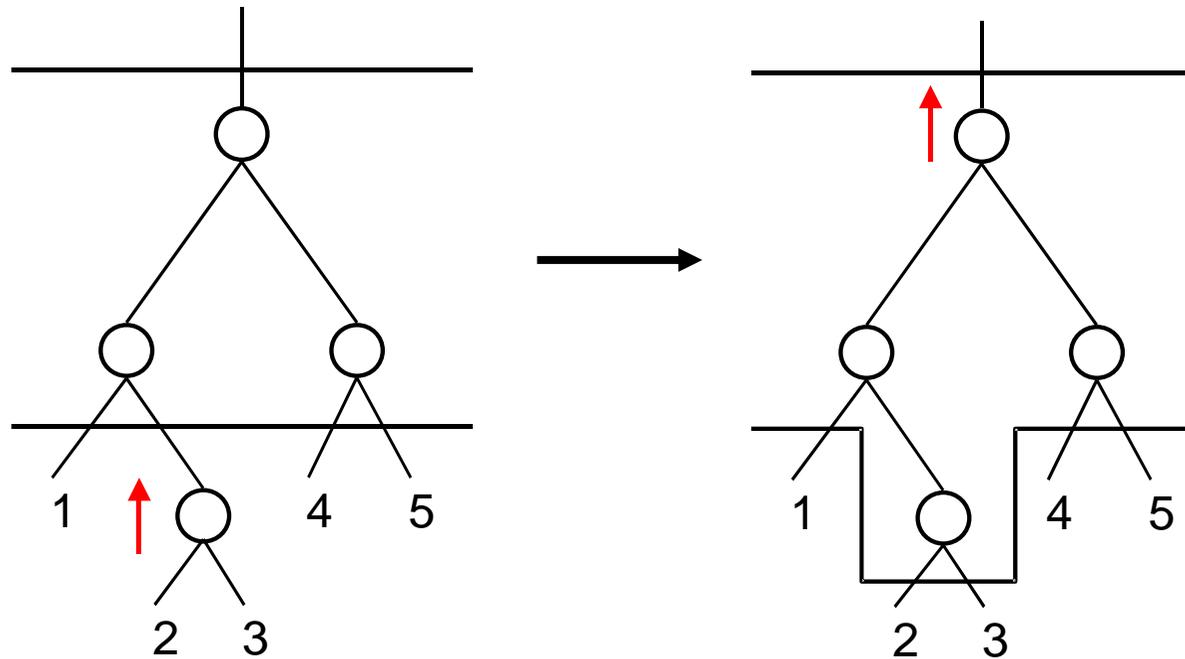
- Push-up-request führt entweder zu einer **lokalen strukturellen Änderung** mit konstantem Aufwand (Fall 1)
- oder zu einer Verschiebung des Push-up-requests auf das nächst höhere Niveau **ohne Strukturänderung** (Fall 2)

Fall1 [Genug Platz auf nächsthöherer Schicht]



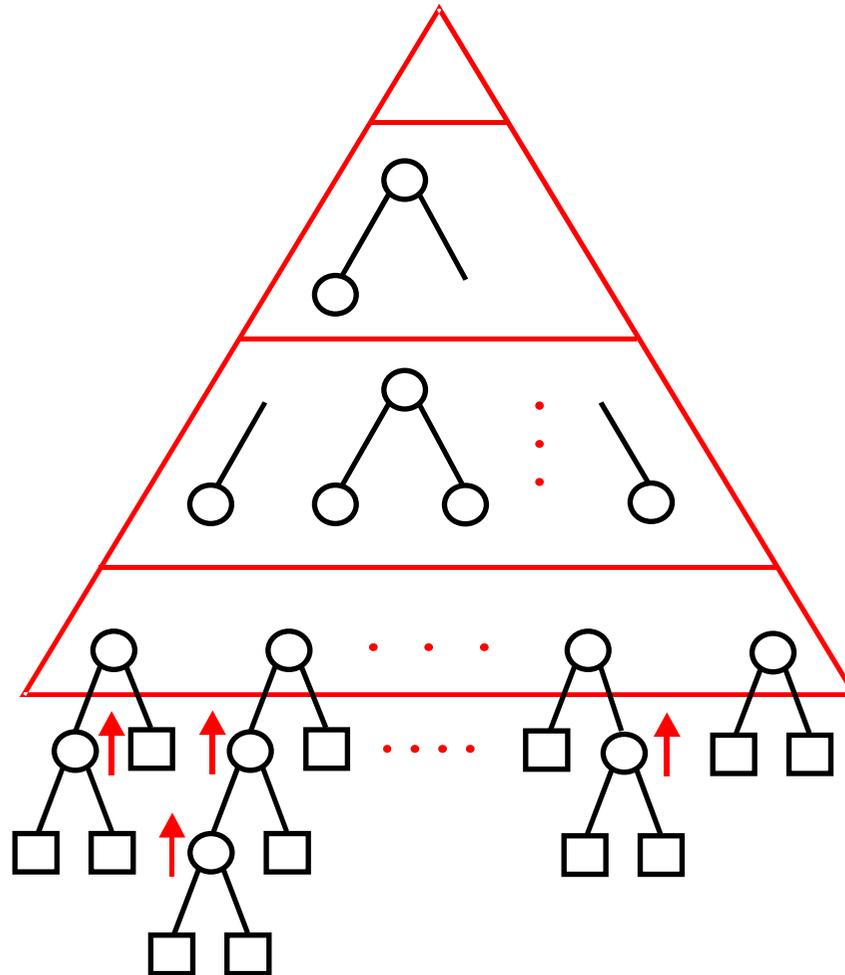
Beseitigung eines Push-up-requests (2)

- Fall 2 [Nächste Schicht voll]



Schaffe ggfs. neue Schicht an der Spitze

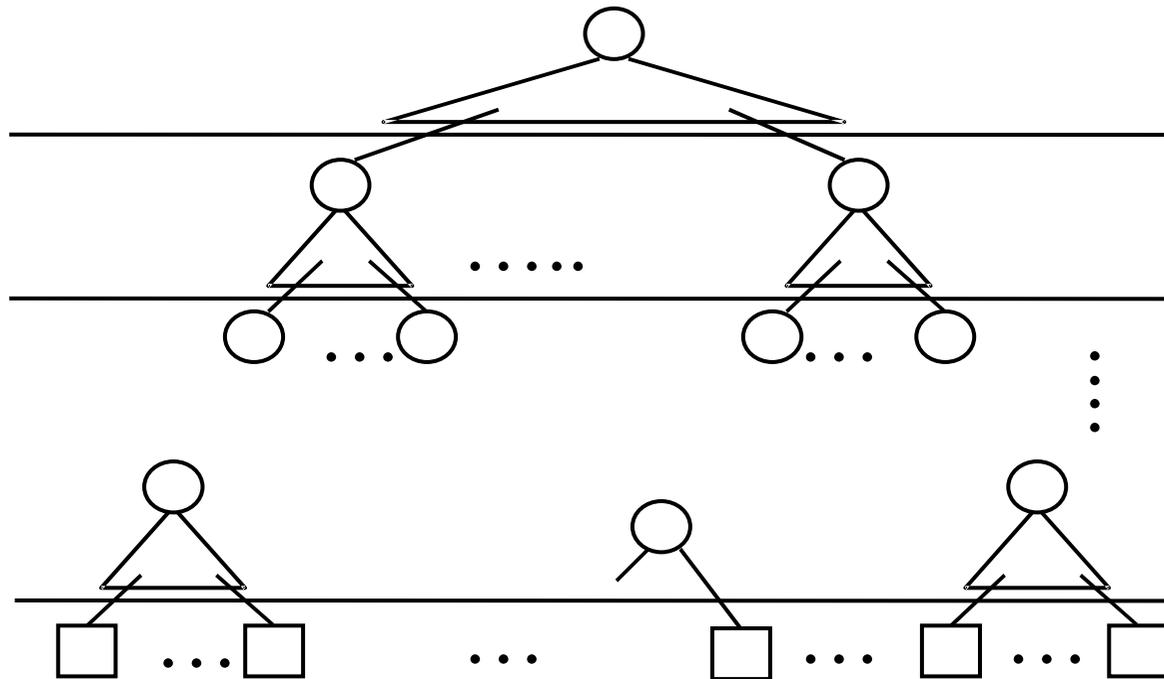
Beseitigung mehrerer Push-up-Requests



Unterhalb unterster Straßengrenze:
Von oben nach unten!

Sonst beliebig nebenläufig!

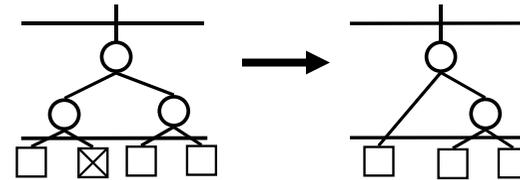
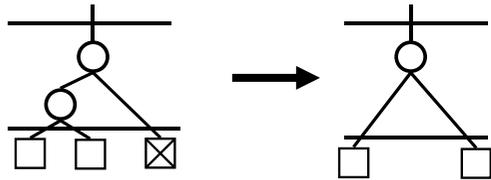
Entfernen



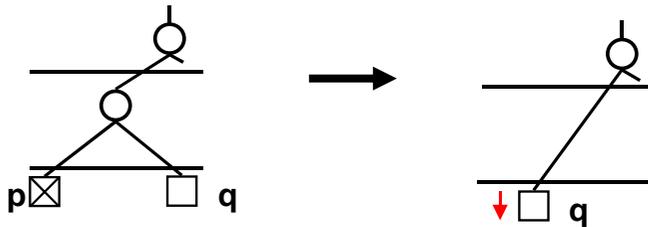
Deponiere für jede Entfernung einen **Löschvermerk**.
Beseitige Löschvermerke.

Beseitigung eines Löschermerks

Fall 1 [genügend Knoten auf unterster Schicht]



Fall 2 [unterste Schicht zu dünn]



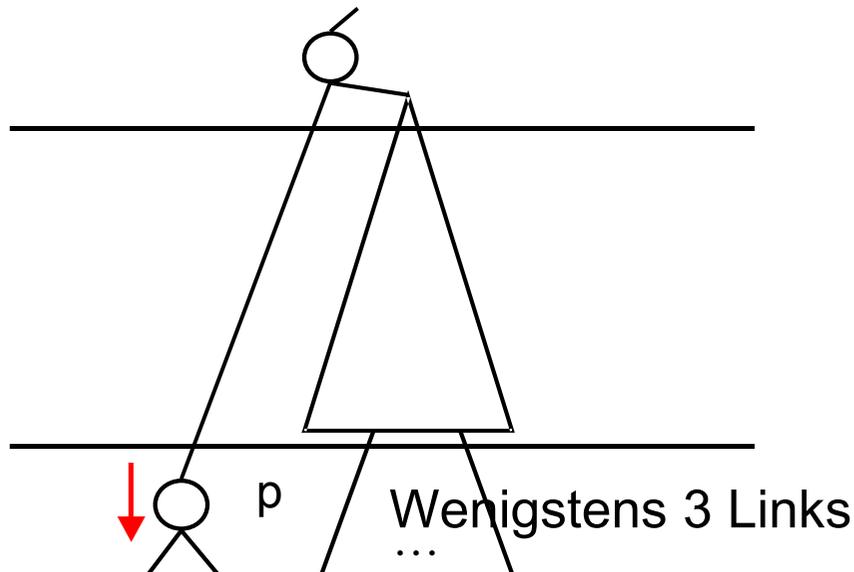
Deponiere „pull-down-request“ ↓

Beseitigung eines Pull-down-requests

Behandlung eines Pull-down Requests führt

- entweder zu einer Strukturänderung (mit konstanten Kosten) und Halt,
- oder (ausschließlich) zu einer rekursiven Verschiebung auf nächste Schicht ohne Strukturänderung

Fall 1 [es gibt “genügend” Knoten auf nächsthöherer Schicht]

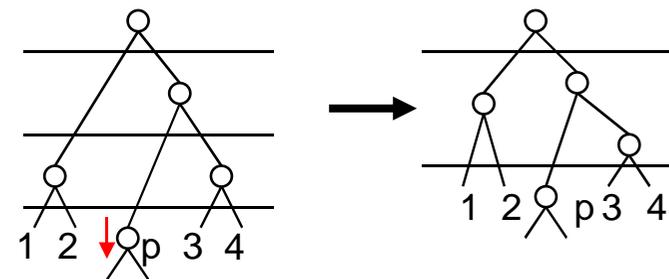
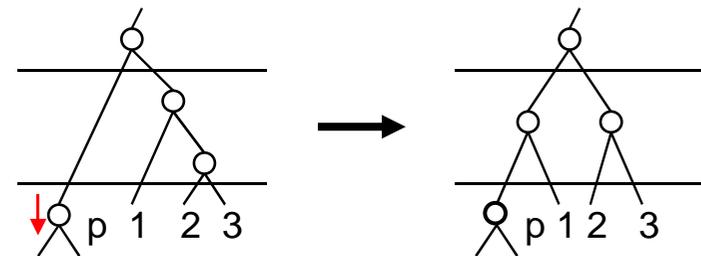
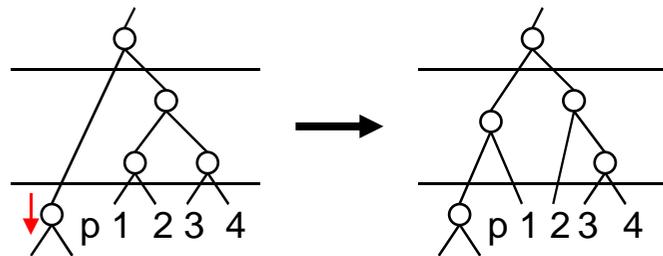


Umstrukturieren und Halt!

Beseitigung von Pull-down-requests (1)

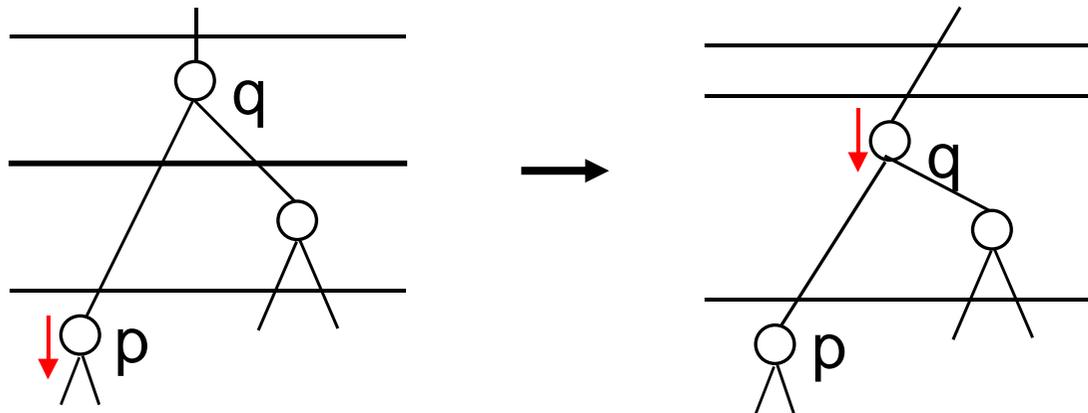
Fall1 [genügend Knoten auf nächster Schicht]

Strukturänderung konstanten Aufwands und Halt!



Beseitigung von Pull-down-requests (2)

Fall2 [nicht genügend Knoten auf nächster Schicht]

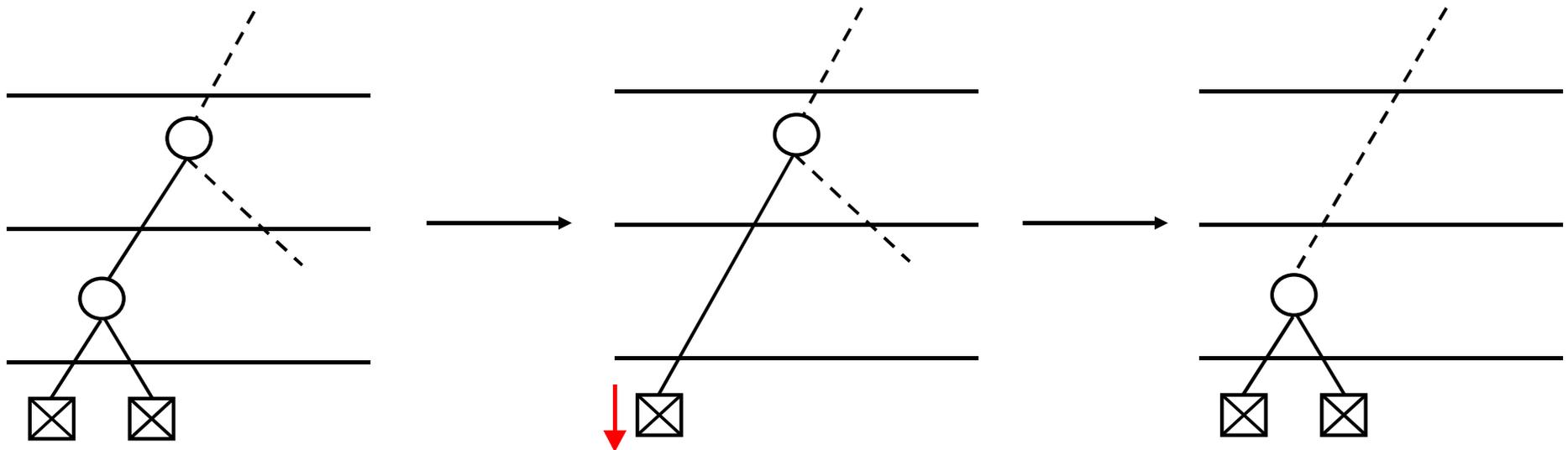


Verschiebe Pull-down-request auf nächst höhere Schicht,
ohne strukturelle Änderungen

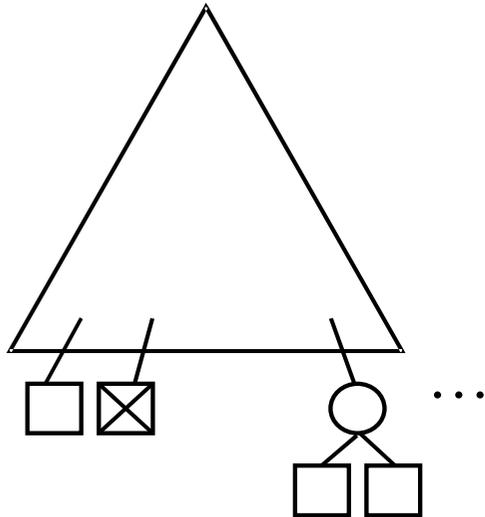
(Beseitige ↓ und oberste Schicht, wenn q Wurzel ist.)

Folgen von Entferne-Operationen

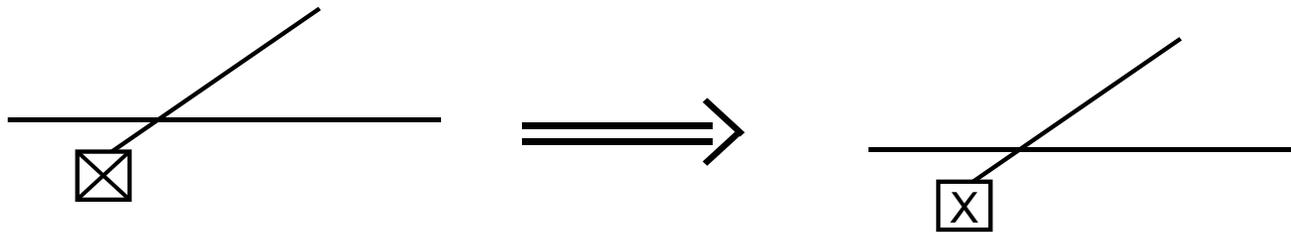
- Deponiere eine Entferne-Markie für jede Delete-Operation
- Beseitige die Entferne-Markie in beliebiger Reihenfolge, aber so, dass sie sich nicht gegenseitig stören.



Nebenläufige Updates und Rebalancierungen

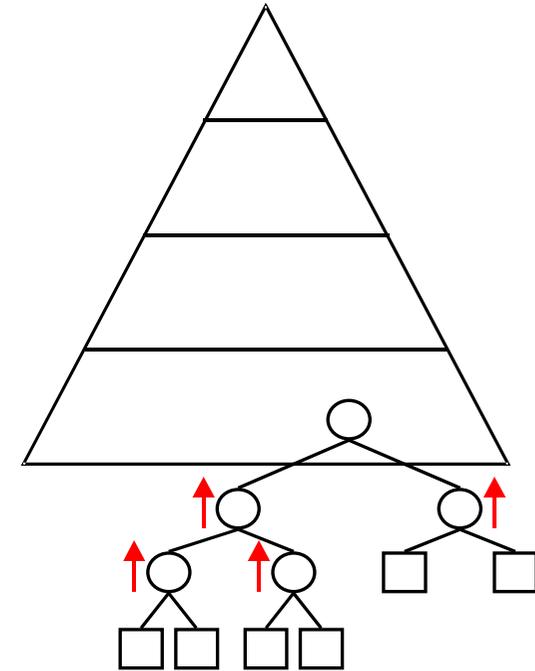
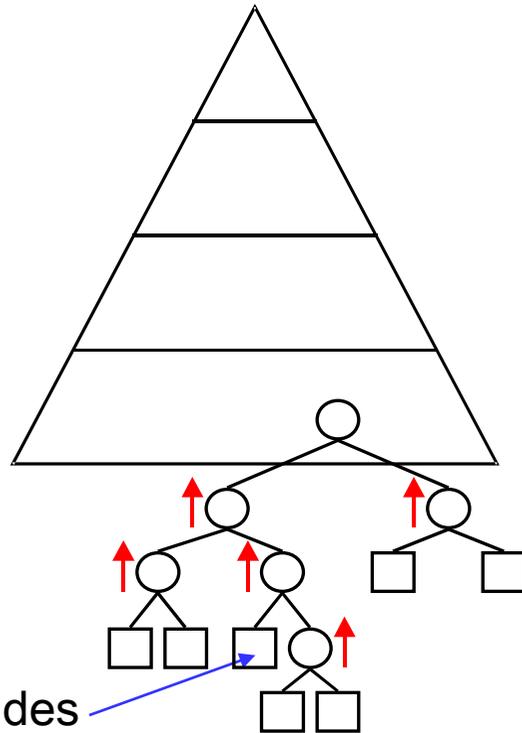


Updates betreffen verschiedene Blätter



Einfügen eines Schlüssels x
In einem Blatt mit Löschrunde

Nebenläufige Updates und Rebalancierungen



Zu entfernendes
Blatt

Behandle Einfügungen, Entfernungen, \uparrow , \downarrow nebenläufig in beliebiger Reihenfolge
Nach dem Prinzip „**Topmost first**“ und **vermeide Überlappungen** von Transformationen!

Schlussbemerkung (1)

- Viele mögliche Strategien für die nebenläufige Behandlung von   
- Reduktion und Zusammenfassung von verschiedenen Fällen bei der Beseitigung von   
- Erhöhung der Effizienz durch Zusammenfassung verschiedenen Umstrukturierungen möglich.

Schlussbemerkung (2)

