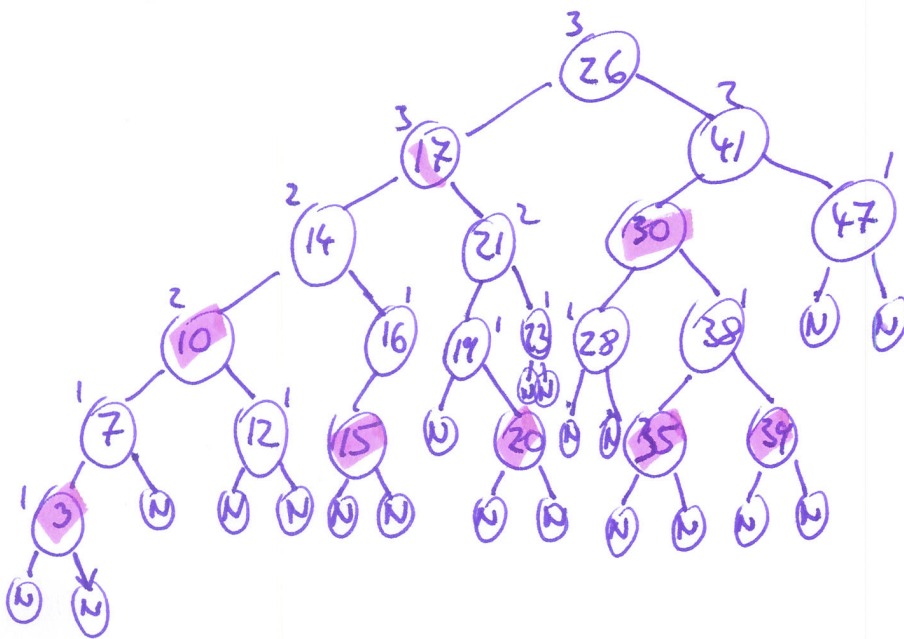


## 4.6 Andere Baumstrukturen

### 4.6.1 Rot-Schwarz-Bäume

Eigenschaften

0. Binärer Suchbaum, alle Blätter sind "NIL"



Kein AVL-Baum!

1. Jeder Knoten ist entweder rot oder schwarz
2. Die Wurzel ist schwarz.
3. Jedes Blatt (NIL) ist schwarz.
4. Wenn ein Knoten rot ist, dann sind seine beiden Kinder schwarz
5. Für jeden Knoten enthalten alle Pfade, die an einem Knoten starten und in einem Blatt des Teilbaumes dieses Knotens enden, die gleiche Anzahl schwarzer Knoten.

Man kann folgende Dinge beweisen:

100

### Satz 4.12

Ein Rot-Schwarz-Baum mit  $n$  inneren Knoten hat höchstens die Höhe  $2 \log(n+1)$ .

Beweis: Nicht hier!

### Satz 4.13

Löschen und Einfügen (samt notwendiger Reparaturoperationen zum Erhalt der Eigenschaften 0.-5.) lassen sich in einem Rot-Schwarz-Baum in  $O(\log n)$  durchführen, wobei  $n$  die Zahl der inneren Knoten ist

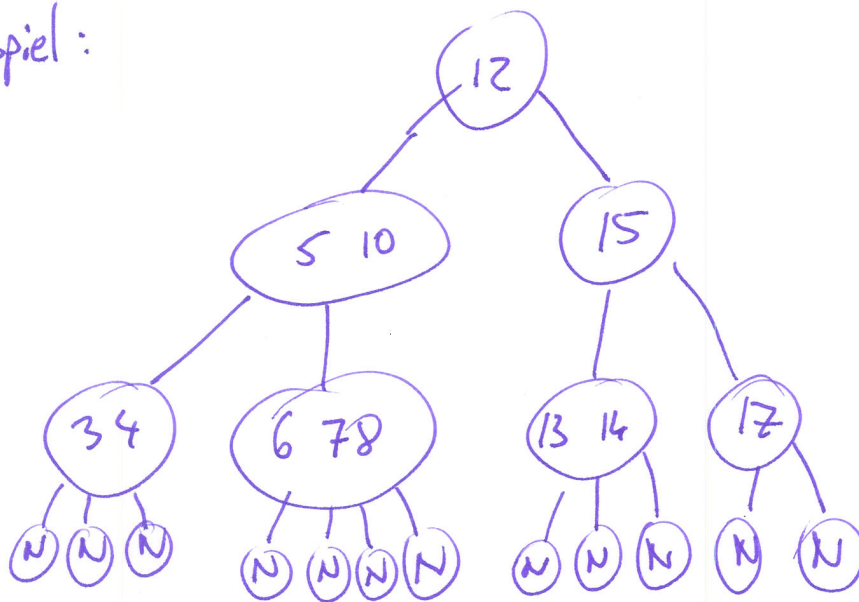
Historisch:

Bayer (1972) - Grundidee

Guibas (Sedgwick (1978)) - Farben + Eigenschaften

- Grundideen:
- (I) Erlaube mehr als ein Objekt pro Knoten (z.B. zwei oder drei)
  - (II) Erlaube mehr als zwei Kinder (z.B. bis zu vier)

Beispiel:



Eigenschaften:

- (i) Jeder Knoten hat höchstens 3 Objekte,
- (ii) Jeder Knoten hat höchstens 4 Kinder.

Sowie

- (iii) Jeder innere Knoten hat mindestens zwei Kinder
- (iv) Alle Blätter haben dieselbe Tiefe

### Satz 4.14

Die Höhe eines  $(2,4)$ -Baums für  $n$  Objekte ist  $\Theta(\log n)$ .

### Satz 4.15

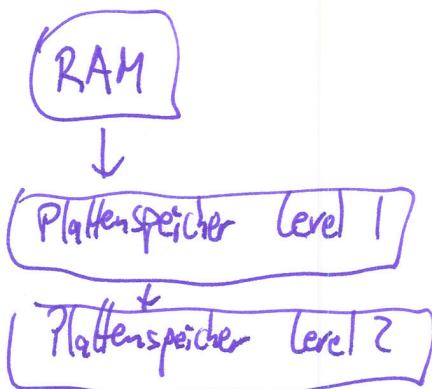
Einfügen und Löschen in einem  $(2,4)$ -Baum für  $n$  Objekte (samt Reparatur) ist in  $O(\log n)$  möglich.

Beweis: Nicht hier!

### 4.6.3 B-Bäume

Idee: Speicherhierarchien!

("Cache")

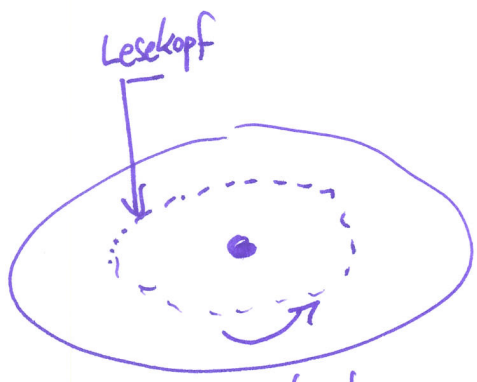


Geschwindigkeit des Zugriffs:

RAM : schnell

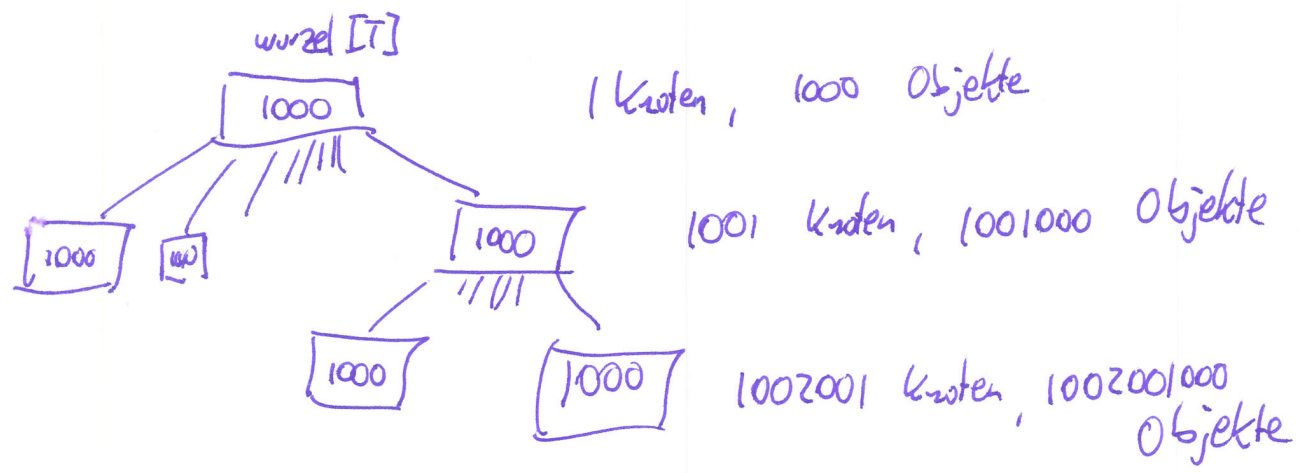
Platte : Fixkosten (richtige Stelle auf Platte finden),  
dann moderate Zugriffs-kosten pro Block.  
Speicherzellen

→ Idee: Bei Zugriff gleich mehr Daten „schauteln“!



rotiert - eventuell ganzer Umlauf erforderlich!

Konsequent umgesetzt:



→ B-Bäume!

mehr nicht hier, siehe Literaturhinweise



Historisch + Literatur :

- Karth 1973
- Alvo Hopcraft, Ullman 1974
- Sedgwick 1988

Aktuelle: Forschung:

Speichergröße nicht immer bekannt, man hat nicht mehr nur starre Speicherhierarchien.

Bender, Demire, Farach-Colton (2000):  
(\*1970) (\*1981) (\*1964)

"Cache-oblivious B-trees"

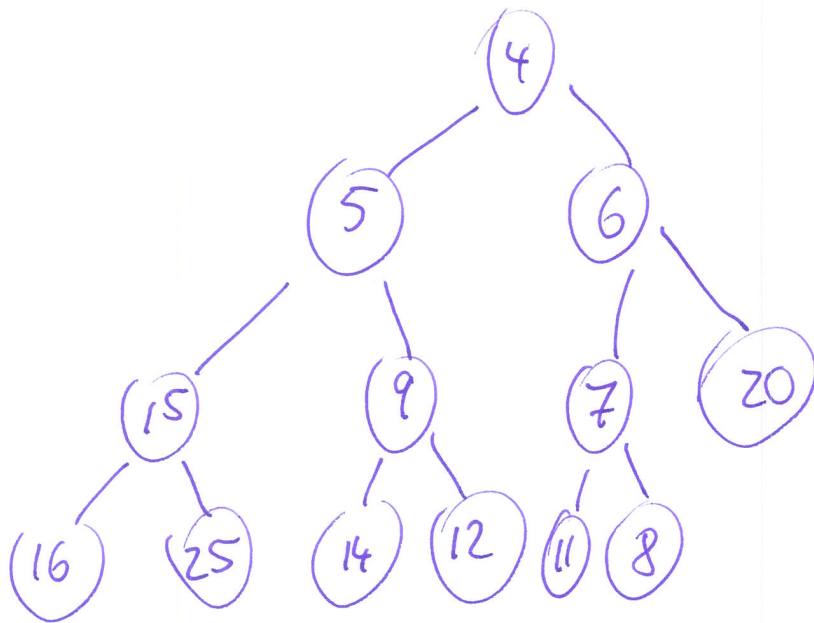
Startup-Firma "Tokutek"

"Tokutek was founded to dramatically enhance the performance of databases and file systems. Tokutek's software is based on eight years of research in cache-oblivious algorithms and will dramatically accelerate key database and file system operations."

↳ Links!

## 4.6.4 Heaps („Haufen“)

Binärbaum, aber mit anderem Ordnungsprinzip!



Der ~~Objekt~~ <sub>wert</sub> in jedem Knoten ist ~~größer~~ kleiner als in jedem Kind!

↳ Verschiedene Operationen, nicht hier!