Satz 4.14

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

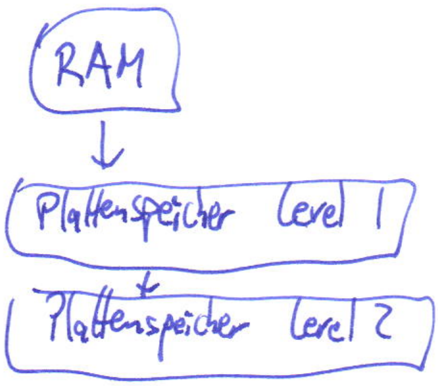
Satz 4.15

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

Beweise: Nicht hier!

46.3 B-Bäume

Idee: Speicherhierarchien! ("Cache")



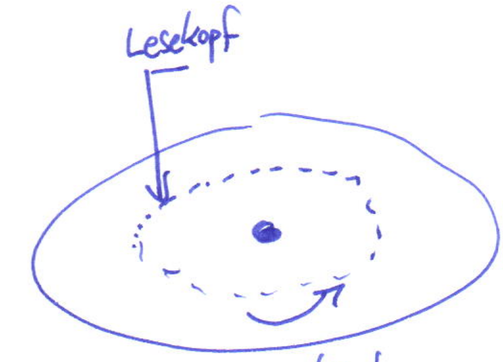
Geschwindigkeit des Zugriffs:

RAM : schnell

Platte : Fixkosten (richtige Stelle auf Platte finden),

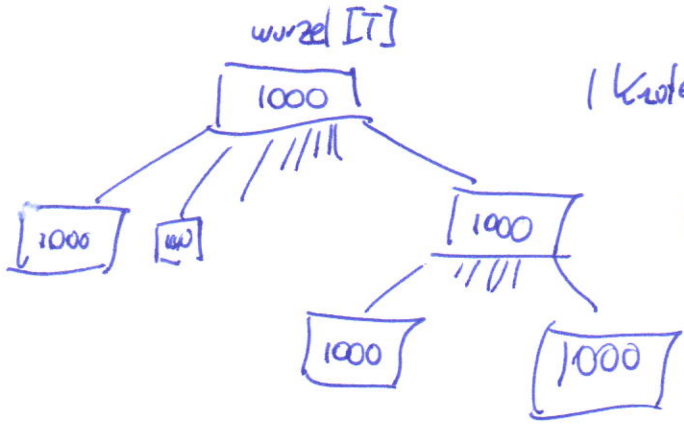
dann moderate Zugriffskosten pro Block.
Speicherzellen

→ Idee: Bei Zugriff gleich mehr Daten „schaufeln"!



rotiert - eventuell ganzer Umlauf erforderlich!

Konsequent umgesetzt:



1 Knoten, 1000 Objekte

1001 Knoten, 1001000 Objekte

1002001 Knoten, 1002001000 Objekte

→ B-Bäume!

Mehr nicht hier, siehe Literaturhinweise

Historische Literatur:

Knuth 1973

Aho, Hopcroft, Ullman 1974

Sedgwick 1988

Aktuelle Forschung:

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

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

„Cache-oblivious B-trees"

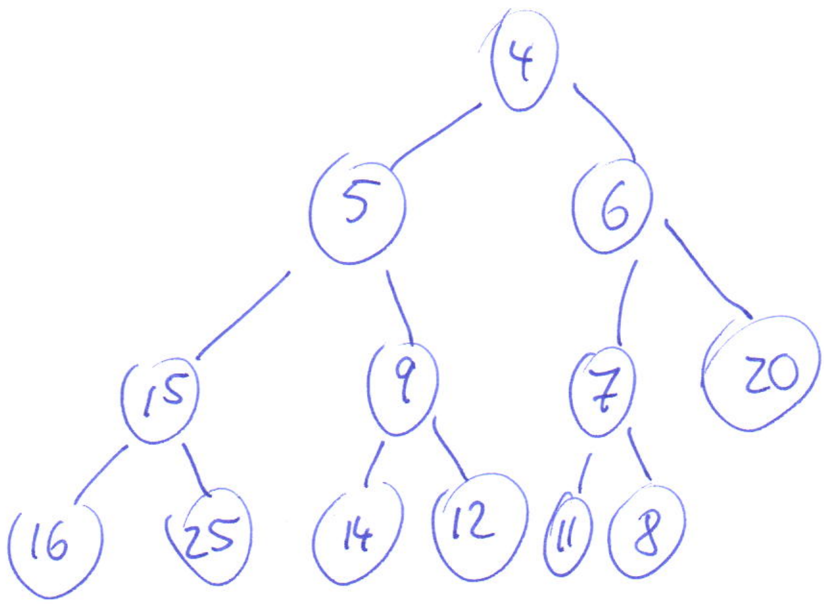
Startup-Firma „Tokutek"

"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 algorithmics and will dramatically accelerate key database and file system operations."

↪ Links!

4.6.4 Heaps („Haufen")

Binärbaum, aber mit anderem Ordnungsprinzip:



Der ~~Objekt~~ Wert in jedem Knoten ist ~~größer~~ kleiner als in jedem Kind!

↳ Verschiedene Operationen, nicht hier!

Kapitel 5: Sortieren

5.0 Vorspann: Sortieren von Übungszetteln

5.1 Sortieren und Permutationen

Gegeben: Eine Menge von Objekten $x_1, \dots, x_n$,
eine Größenrelation „<", die
je zwei Objekte ordnet
(→ Totalordnung!)

Gesucht: Eine Sortierung der Objekte nach
Reihenfolge.

Bemerkungen:

(1) Wir schreiben der Einfachheit halber
$$x_i < x_j$$
bzw $x_j < x_i$
(und gehen wie gesagt davon aus,
das eines davon gilt)

(2) Statt die Objekte in der vorgegebenen
Reihenfolge hinzuschreiben, reicht es auch,
die Permutation zu kennen, die
die Anordnung beschreibt:

2 4 3 1 5

Bauer Mejer Konrad Anton Maier

↳ 1 2 3 4 5

Anton Bauer Konrad Maier Mejer

Also hat man Zugriff in sortierter Form durch Codierung

4 1 3 2 5

(Verwirrend? Konzentrationssache!)

Schreibweise für Permutationen:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 2 & 5 \end{pmatrix}$$

~~Bedeutung~~: Interpretation als Pointer

Größtes Objekt an Stelle 4
Zweitgrößtes Objekt an Stelle 1
Drittgrößtes Objekt an Stelle 3
Viertgrößtes Objekt an Stelle 2
Fünftgrößtes Objekt an Stelle 5

Interpretation als Transportanleitung: d.h. ~~[illegible]~~

Bringe Objekt aus 1 nach 4

aus 2 nach 1

aus 3 nach 3

aus 4 nach 2

aus 5 nach 5 .

Hintereinanderausführung:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 2 & 5 \end{pmatrix} \quad \text{gefolgt von} \quad \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

also:

Objekt aus 1 nach 4 nach 2

aus 2 nach 1 nach 5

aus 3 nach 3 nach 3

aus 4 nach 2 nach 4

aus 5 nach 5 nach 1

$$\to \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 3 & 4 & 1 \end{pmatrix}$$

Sortieren als Transportvorgang:

Welche Permutation macht

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 2 & 5 \end{pmatrix}$$ rückgängig?

1 nach 4 – nach 1

2 nach 1 – nach 2

3 nach 3 – nach 3

4 nach 2 – nach 4

5 nach 5 – nach 5

Also $$\begin{pmatrix} 4 & 1 & 3 & 2 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

bzw. (sortiert!)

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 3 & 1 & 5 \end{pmatrix}$$

Also: – Zugriff in aktueller **Anordnung** in sortierter Form per Pointer:

Permutation der Objekte, also **Anordnungspermutation**

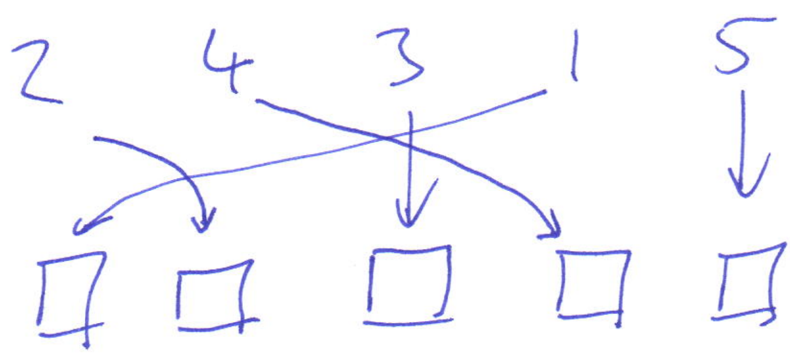
– Rückgängimachen der Unordnung, um die Objekte explizit in sortierte Reihenfolge zu bringen:

Inverse Permutation der Objekte, d.h. **Rücksortierungspermutation**

5.2 Vorbetrachtungen zum Modell

Beim Sortieren gibt es wichtige Unterschiede!

(1) Kennt man explizit die Nummerierung der Objekte, dann geht es schnell:



← Also

~~(2) Kennt man keine explizite Nummerierung, sondern kann die Objekte nur paarweise vergleichen (Balkenwaage!), so wird es schwerer!~~

Beobachtungen:

(i) Schaut man sich bei Nummerierung ein Objekt an, reduziert sich die Zahl der Möglichkeiten für die Anordnungspermutation drastisch!

(Schaut man sich das erste Objekt an und identifiziert eine 2, so bleiben von $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ möglichen Anordnungen nur noch 24 Kandidaten übrig, danach noch 6, etc.!)