Bulk Loading-Methode

Sascha Szott

Fachgebiet Informationssysteme
HPI Potsdam

19. Mai 2008

Problemstellung

- Menge von r Records soll mittels B^+-Baum als Indexstruktur indiziert werden
- naiver Ansatz: führe ausgehend von einem leeren B^+-Baum nacheinander r Einfügeoperationen aus
  - jede Einfügeoperation erfordert Suchoperation (Laufzeit)
  - noch schlimmer: ist ein Knoten (Block) nicht im Hauptspeicher, ist teurer Festplattenzugriff notwendig (I/O-Kosten)
- Abhilfe: *Bulk Loading*

Bulk Loading

1. sortiere die r Schlüssel-Zeiger-Paare aufsteigend nach dem Schlüsselwert (Schlüsselwert ist nur ein Bruchteil des Records)
2. allokiere leeren Block als Wurzel und füge Zeiger auf die ersten zwei Blöcke (in Sortierreihenfolge) ein
3. füge nun für jeden weiteren Block Schlüssel-Zeiger-Paar in Wurzel hinzu bis diese voll ist
4. wenn Wurzel voll: spalte diese auf und erzeuge neue Wurzel $\rightarrow$ verschiedene Aufspaltungsvarianten
   - gleichmäßige Aufteilung (Füllstand $\approx 50\%$)
   - falls noch ausreichend viele Blöcke vorhanden: erzeuge leeren Knoten (Füllstand $\approx 100\%$)
   - ...
5. füge für jeden weiteren Block Schlüssel-Zeiger-Paar in den inneren Knoten, der am weitesten rechts auf vorletzter Stufe des Baumes liegt (falls voll: Aufspaltung wie oben)

Füllstand der Knoten

abhängig vom Anwendungsfall

- „statische“ Relation: möglichst hoch $\rightarrow$ Einsparung von Speicherplatz; wahrscheinlicher, dass Index länger im Hauptspeicher verweilt
- „dynamische“ Relation:
  - häufige Einfügeoperationen: geringer Füllstand $\rightarrow$ im Schnitt weniger Aufspaltungen von Knoten erforderlich
  - häufige Löschoperationen: hoher Füllstand $\rightarrow$ im Schnitt weniger Zusammenfügungen von Knoten erforderlich

Beispiel

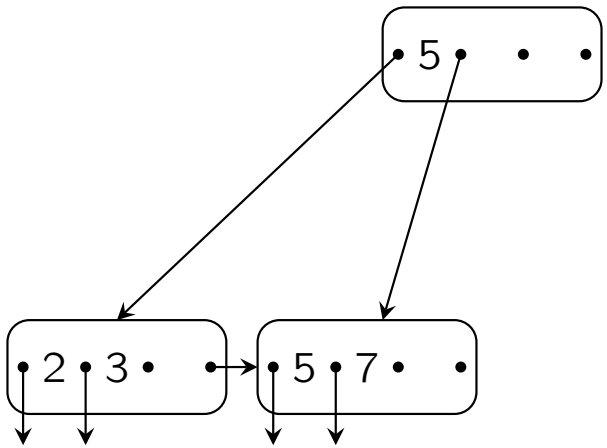
- im Folgenden sei $n = 3$
  - Blätter haben $\left\lfloor \frac{3+1}{2} \right\rfloor = 2$ oder 3 Zeiger auf Records (im Fall eines dichtbesetzten Index)
  - innere Knoten haben $\left\lceil \frac{3+1}{2} \right\rceil = 2$, 3 oder 4 Zeiger auf Knoten der nächst tieferen Stufe
- gleichmäßige Aufspaltungsstrategie
- Menge der einzufügenden Suchschlüssel: alle Primzahlen zwischen 2 und 47 (jeweils inklusive) $\rightarrow$ Sortierung schon vorgenommen

Beispiel

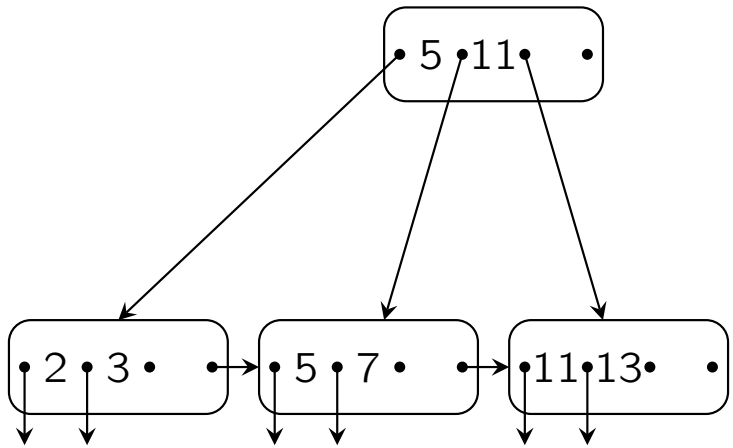
erzeuge jeweils halbgefüllte Blattknoten

- $(2, 3)$
- $(5, 7)$
- $(11, 13)$
- $(17, 19)$
- $(23, 29)$
- $(31, 37)$
- $(41, 43, 47)$ (beachte: Blattknoten muss mindestens 2 Schlüssel-Zeiger-Paare enthalten)

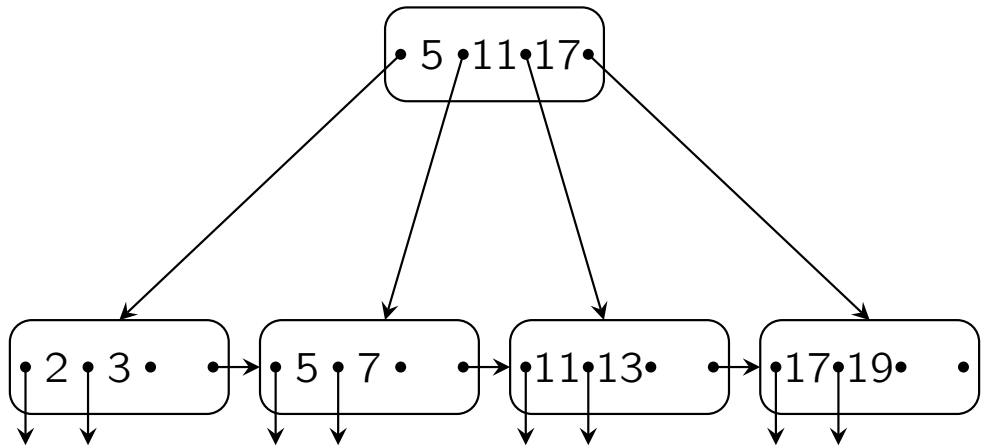
erzeuge Wurzel mit Zeigern auf die ersten beiden Blätter



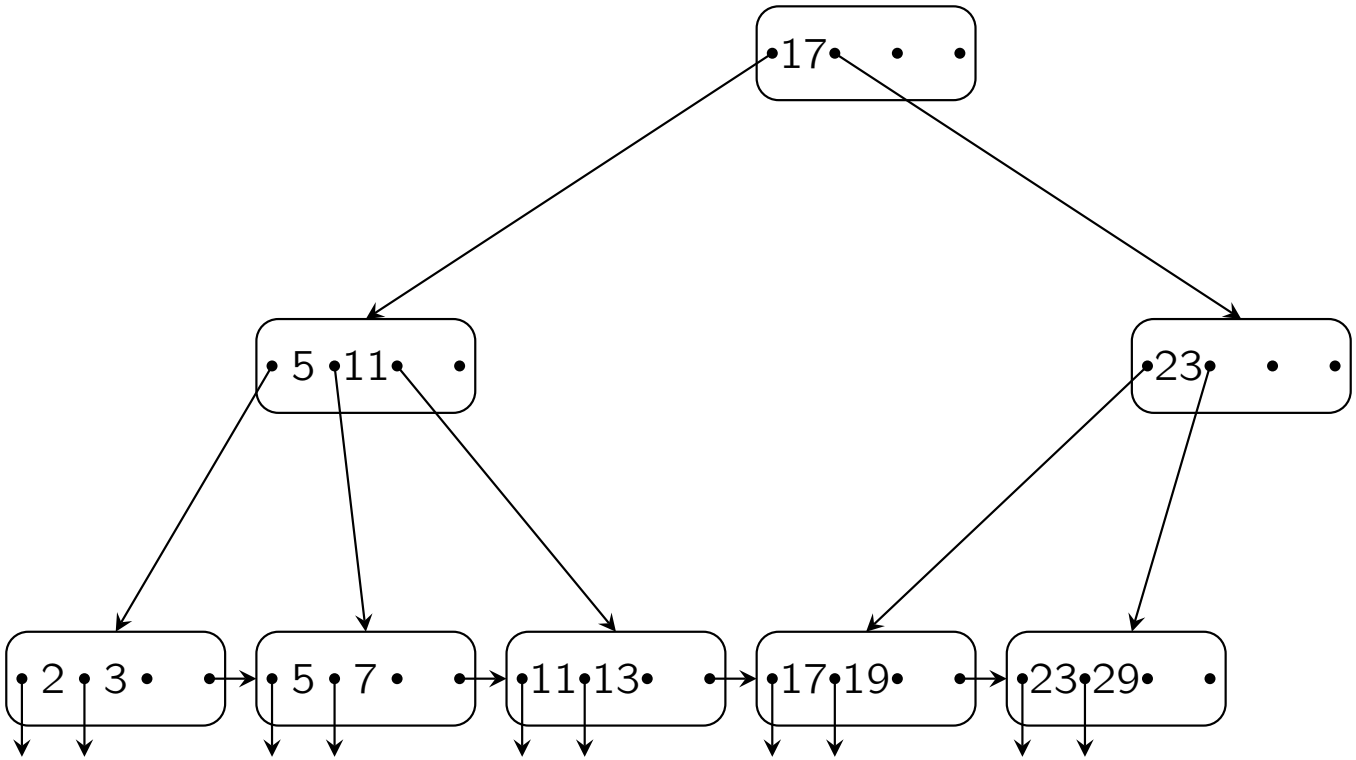
füge Blatt (11, 13) ein



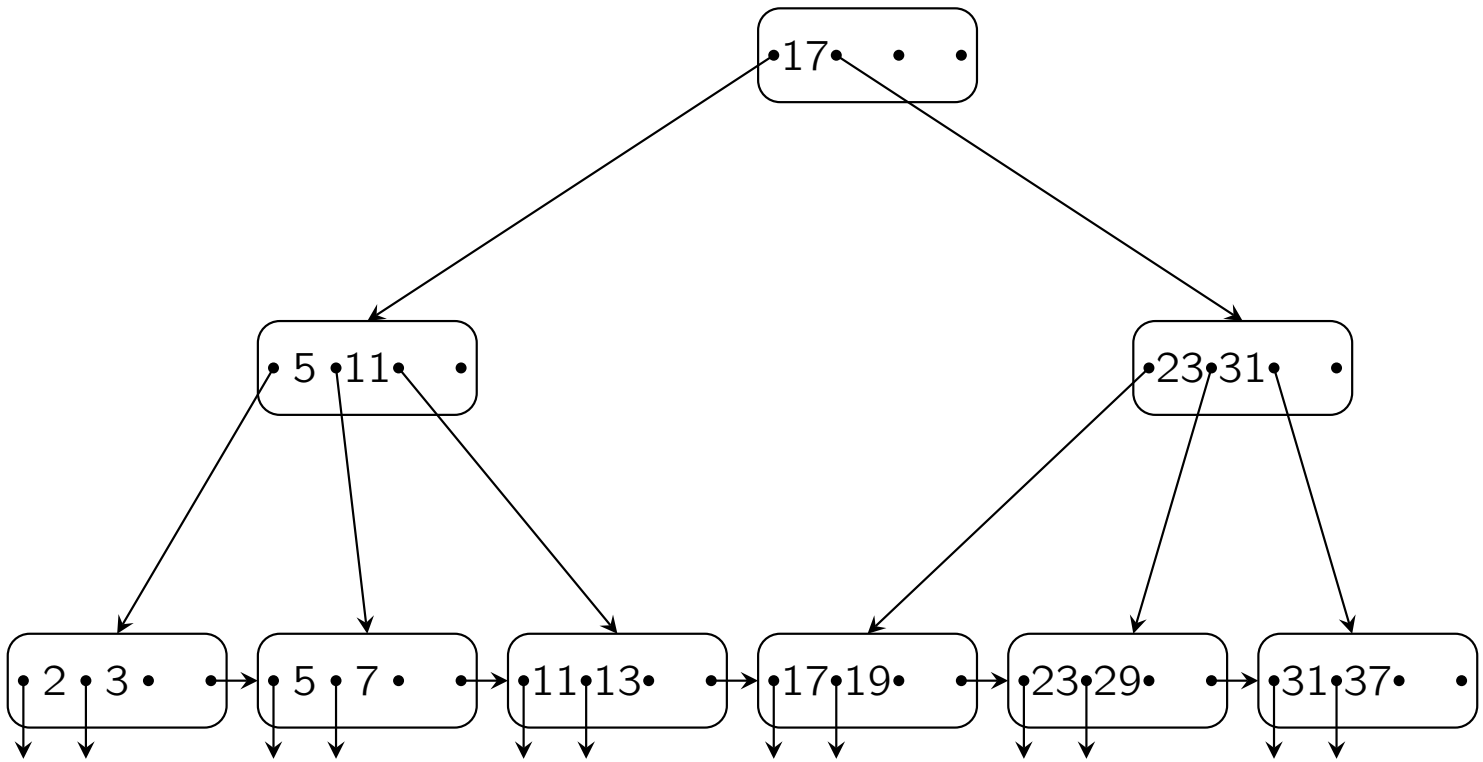
füge Blatt (17, 19) ein



füge Blatt $(23, 29)$ ein



füge Blatt $(31, 37)$ ein



füge Blatt $(41, 43, 47)$ ein