Wiederholung

Definition: B-Baum der Ordnung $\mathbf{m}$ (Bayer und McCreight (1972))

- Jeder Knoten enthält höchstens $2m$ Schlüssel.
- Jeder Knoten außer der Wurzel enthält mindestens m Schlüssel.
- Die Wurzel enthält mindestens einen Schlüssel.
- Ein Knoten mit k Schlüsseln hat genau $k + 1$ Söhne.
- Alle Blätter befinden sich auf demselben Level.

Einfügen in einen B-Baum: Suche Knoten B, in den ein Schlüssel k eingeordnet werden würde.

- **Fall 1**: B enthält $\leq 2m$ Schlüssel
  $\Rightarrow$ füge k in B ein
- **Fall 2**: B enthält $2m$ Schlüssel
  $\Rightarrow$ Overflow-Behandlung: Split des Blattknotens (kann sich bis zur Wurzel auswirken)

Löschen aus einem B-Baum: Suche zu löschenden Schlüssel k.

- **Fall 1**: k befindet sich in einem Blattknoten B
  - **Fall 1.1**: B hat noch mehr als m Schlüssel
    $\Rightarrow$ lösche Schlüssel
  - **Fall 1.2**: B hat genau m Schlüssel
    $\Rightarrow$ Underflow-Behandlung: Betrachte Bruderknoten (immer den rechten falls vorhanden)
    - **Fall 1.2.1**: Bruder hat mehr als m Knoten
      $\Rightarrow$ ausgleichen mit Bruder
    - **Fall 1.2.2**: Bruder hat genau m Knoten
      $\Rightarrow$ verschmelzen mit Bruder (kann sich bis zur Wurzel auswirken)
- **Fall 2**: k befindet sich in einem inneren Knoten B
  $\Rightarrow$ vertausche Schlüssel mit dem größten Schlüssel im linken Teilbaum (d.h. Rückführung auf Fall 1)

Hash-Verfahren

Idee: Verwende Funktion, die aus den Schlüsseln K die Seitenadresse A berechnet (*Hash-Funktion*)

- Verfahren mit Directory: Erweiterbares Hashing
  - Hashfunktion: $h(k)$ liefert Bitfolge $(b_1, b_2, ..., b_d, ...)$
  - Directory besteht aus eindimensionalen Array $D\left[0...2^d - 1\right]$ aus Seitenadressen. d heisst Tiefe des Directory.
  - Verschiedene Einträge können auf die gleiche Seite zeigen.
- Verfahren ohne Directory: Lineares Hashing
  - Hashfunktion: $h(k)$ liefert direkt eine Seitenadresse
  - Problem: Was ist, wenn Datenseite voll ist?
  - Lösung: Überlaufseiten werden angehängt. Bei zu vielen Überlaufseiten degeneriert die Suchzeit.
  - Dynamisches Wachstum der Primärdatei
  - Folge von Hashfunktionen: $h_0, h_1, h_2, ...$
  - Erweitern der Primärdatei um jeweils eine Seite
  - Feste Splitreihenfolge, vorgegeben durch Expansionszeiger
  - Kontrollfunktion: *Wann* wird gesplittet? $\rightarrow$ falls Belegungsfaktor einen Schwellwert übersteigt