



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS16/17

Harald Lang, Linnea Passing (gdb@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1617/grundlagen/>

### Blatt Nr. 11

#### Hausaufgabe 1

Bestimmen Sie  $k$  für einen B-Baum, der die folgenden Informationen aller Menschen auf der Erde (ca. 10 Milliarden) enthalten soll: Namen, Land, Stadt, PLZ, Straße und Hausnummer (insgesamt ca. 100 Byte). Dabei ist die Steuernummer eindeutig und 64 Bit lang und wird im B-Baum als Suchschlüssel verwendet. Gehen Sie bei der Berechnung davon aus, dass eine Speicherseite 16KiB groß ist und ein Knoten des B-Baums möglichst genau auf diese Seite passen sollte.

#### Lösung:

Zunächst gibt uns die Information über die zu erwartende Anzahl von Einträgen im B-Baum einen Hinweis auf die Größe eines Verweises in den Speicher. Wir betrachten gängige Architekturen und sehen, dass eine Speicherung auf einer Maschine, deren Adressierbarer Speicher lediglich  $2^{32}$  byte groß ist, nicht ohne weiteres möglich ist. Wir gehen im Verlauf der Aufgabe also von einer 64 bit Architektur aus, bei der ein Zeiger eine Größe von 8 byte hat.

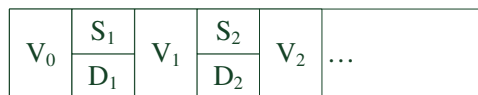


Abbildung 1: Struktur eines B-Baum Knotens

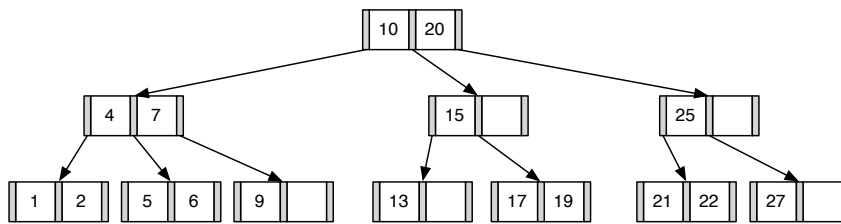
Um nun  $k$  zu bestimmen, muss die von  $k$  abhängige Größe eines Knotens maximiert werden, so dass dieser gerade noch auf eine Seite der Größe 16KiB passt. Die Struktur eines solchen Knotens ist in Abbildung 1 dargestellt. Zu beachten ist, dass ein komplett gefüllter Knoten genau  $2k$  Schlüssel/Daten Paare, jedoch  $2k + 1$  Verweise speichern muss.

Die Berechnung ergibt sich wie folgt:

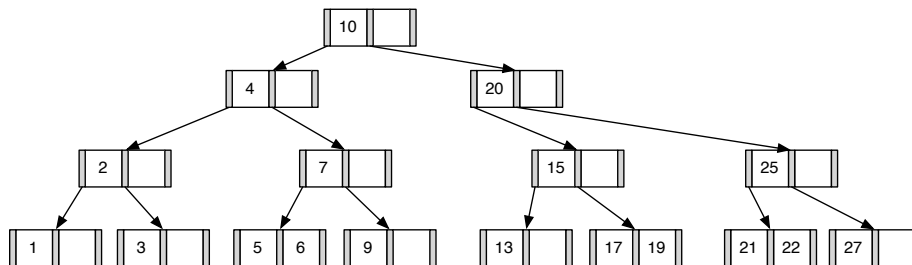
$$\begin{aligned} 2k * (\text{Schlüsselgröße} + \text{Datengröße}) + (2k + 1) * \text{Zeigergröße} &\leq 16\text{KiB} \\ 2k * (8\text{bytes} + 100\text{bytes}) + (2k + 1) * 8\text{bytes} &\leq 16384\text{bytes} \\ 2k * 116\text{bytes} + 8\text{bytes} &\leq 16384\text{bytes} \\ 2k &\leq 16376\text{bytes}/116\text{bytes} \\ k &\leq (16376\text{bytes}/116\text{bytes})/2 \approx 70,59 \end{aligned}$$

Da der Knoten „innerhalb“ einer Speicherseite liegen soll und daher nicht überlappen darf, sollte  $k$  idealerweise auf 70 gesetzt werden, keinesfalls auf 71.

## Hausaufgabe 2

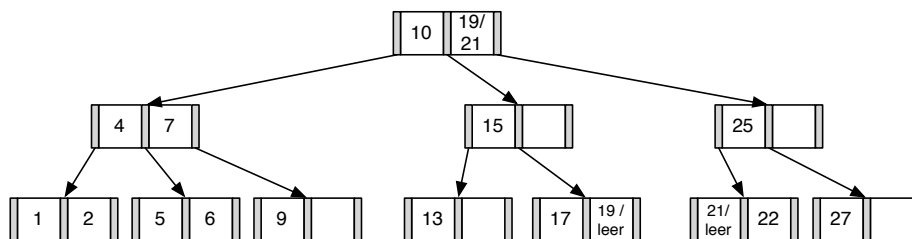


1. Fügen Sie die 3 in den gezeigten B-Baum ein. Zeichnen Sie das Endergebnis. Zeichnen Sie jeweils den kompletten Baum oder machen Sie **deutlich**, falls Teile des Baumes unverändert bleiben. Verwenden Sie den aus der Vorlesung bekannten Algorithmus. Das Ergebnis sieht wie folgt aus:



2. Entfernen Sie aus dem **ursprünglichen Baum** den Eintrag 20. Zeichnen Sie das Ergebnis der Operation. Sollte es mehrere richtige Lösungen geben, genügt es, wenn Sie hier eine angeben. Zeichnen Sie jeweils den kompletten Baum oder machen Sie **deutlich**, falls Teile des Baumes unverändert bleiben. Verwenden Sie den aus der Vorlesung bekannten Algorithmus.

Das Ergebnis sollte wie folgt aussehen:

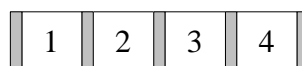


## Hausaufgabe 3

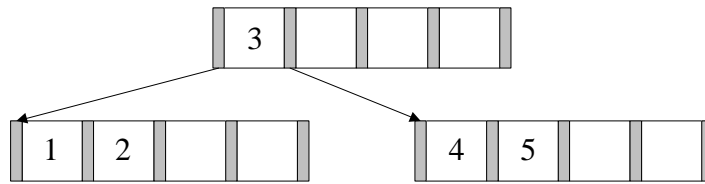
Fügen Sie in einen anfänglich leeren B-Baum mit  $k = 2$  die Zahlen eins bis zwanzig in aufsteigender Reihenfolge ein. Was fällt Ihnen dabei auf?

### Lösung:

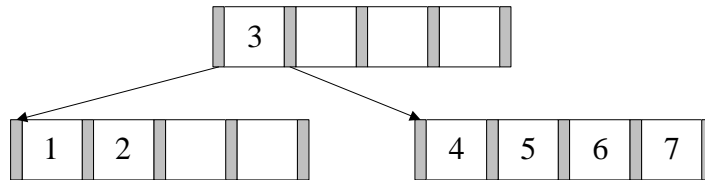
Nachdem man die Zahlen 1 bis 4 eingefügt hat, liegt folgender B-Baum vor:



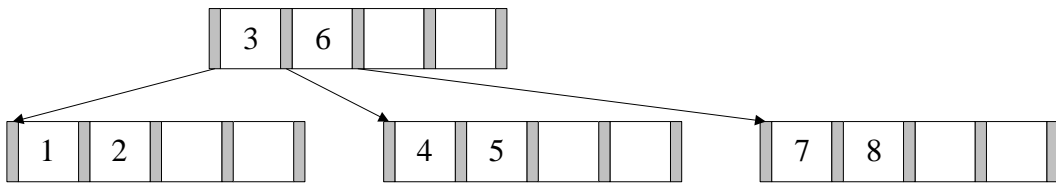
Beim Einfügen von 5 wird der Knoten gespalten und man erhält eine neue Wurzel.



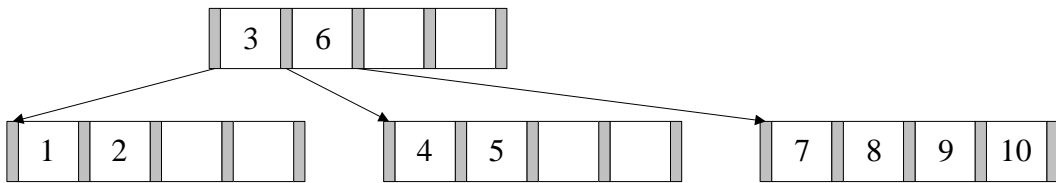
Die nächsten beiden Zahlen lassen sich wieder ohne Probleme einfügen.



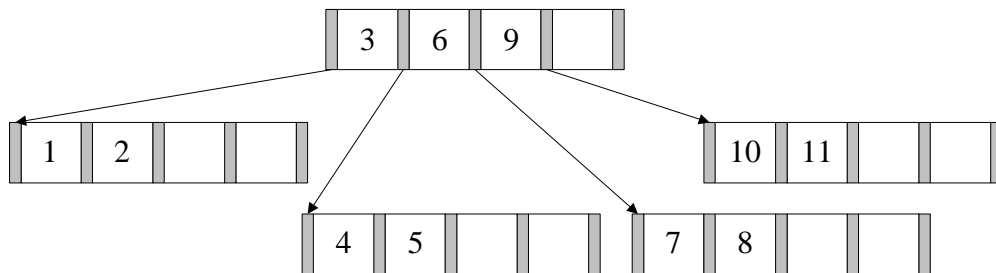
Beim Einfügen der 8 kommt es erneut zum Überlauf. Die 6 wandert in die Wurzel.



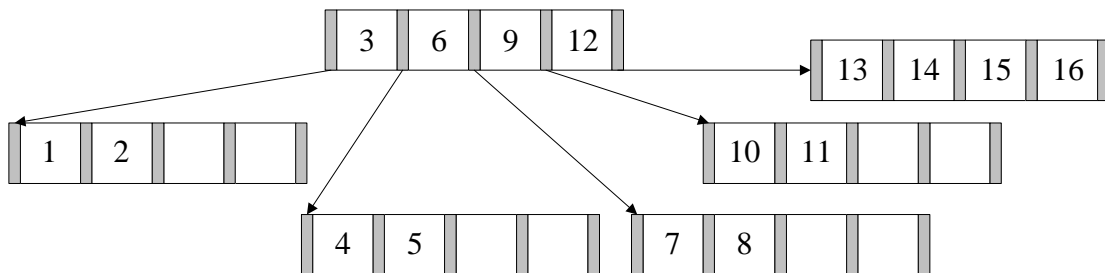
9 und 10 lassen sich wieder ohne Probleme einfügen. Bei 11 kommt es zum Überlauf.



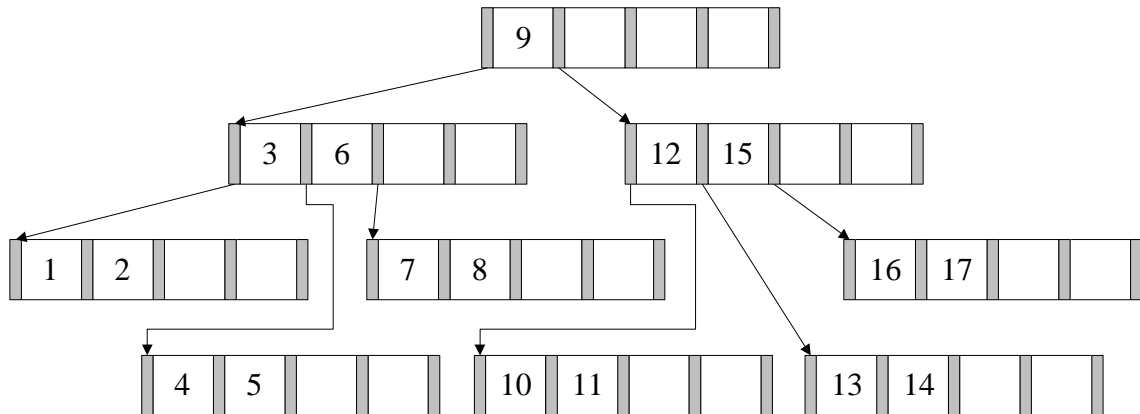
Nach dem Aufspalten erhält man dann:



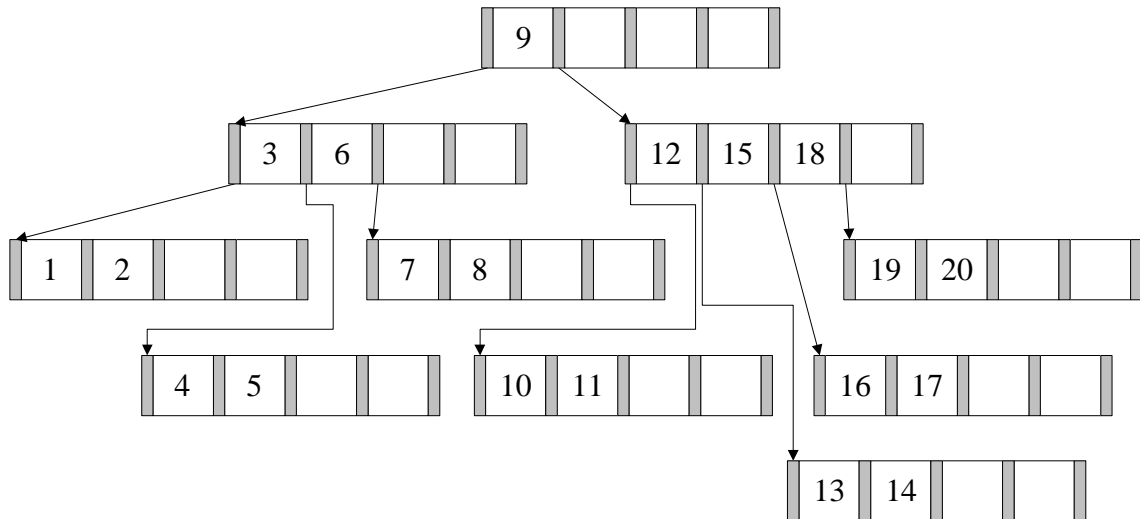
Es werden nun die nächsten Zahlen bis 16 analog eingefügt.



Bei 17 kommt es dann wieder zum Überlauf.



Fügt man nun noch die restlichen Zahlen ein, erhält man folgenden B-Baum:



Es fällt auf, dass der B-Baum nahezu minimale Auslastung aufweist. Dies liegt daran, dass eine aufsteigende Zahlenfolge sequentiell in den Baum eingefügt wird. Nach dem Aufspalten einer Seite in zwei Seiten werden dann in die Seite, die die kleineren Datensätze enthält, keine weiteren Werte mehr eingefügt. Allgemein ist das sortierte Einfügen der Schlüssel in einen B-Baum eine sehr schlechte Idee, da dies zu einer sehr geringen Auslastung führt.

#### Hausaufgabe 4

Geben Sie eine Permutation der Zahlen 1 bis 24 an, so dass beim Einfügen dieser Zahlenfolge in einen (anfangs leeren) B-Baum mit Grad  $k = 2$  ein Baum minimaler Höhe entsteht. Skizzieren Sie den finalen Baum.

- 5,10,15,20 müssen in der Wurzel sein, damit der Baum die Höhe 2 bekommt.  
Z.B.:

- 1,2,5,6,7: 5 ist in Wurzel
- 10,11,12: 5 und 10 sind in Wurzel
- 15,16,17: 1,10 und 15 sind in Wurzel
- 20,21,22: 1,10,15,20 sind in Wurzel
- Ab jetzt übrige Zahlen in beliebiger Reihenfolge einfügen

## Hausaufgabe 5

Betrachten Sie ein abstraktes Relationenschema  $\mathcal{R} = \{A, B, C, D, E, F, G\}$  mit den FDs

$$\begin{aligned}A &\rightarrow BC \\DE &\rightarrow B \\F &\rightarrow A \\E &\rightarrow BF \\A &\rightarrow DE \\C &\rightarrow A.\end{aligned}$$

Überführen Sie die Relation verlustfrei und abhängigkeitsbewahrend in die dritte Normalform.

**Lösung:**

### Kanonische Überdeckung

Linksreduktion:

$$\begin{aligned}A &\rightarrow BC \\~~DE~~ &\rightarrow B \\F &\rightarrow A \\E &\rightarrow BF \\A &\rightarrow DE \\C &\rightarrow A\end{aligned}$$

Rechtsreduktion:

$$\begin{aligned}A &\rightarrow ~~B~~C \\E &\rightarrow \emptyset \\F &\rightarrow A \\E &\rightarrow BF \\A &\rightarrow DE \\C &\rightarrow A\end{aligned}$$

Zusammenfassen der FDs mit gleichen linken Seiten:

$$\begin{aligned}A &\rightarrow CDE \\E &\rightarrow \emptyset \\F &\rightarrow A \\E &\rightarrow BF \\C &\rightarrow A\end{aligned}$$

Entfernen von FDs mit leerer Menge auf der rechten Seite:

$$\begin{aligned}A &\rightarrow CDE \\F &\rightarrow A \\E &\rightarrow BF \\C &\rightarrow A\end{aligned}$$

## Synthesealgorithmus

Zerlegung anhand der kanonischen Überdeckung:

$$\mathcal{R}_1 = \{\underline{A}, C, D, E\}$$

$$\mathcal{R}_2 = \{A, \underline{F}\}$$

$$\mathcal{R}_3 = \{B, \underline{E}, F\}$$

$$\mathcal{R}_4 = \{A, \underline{C}\}$$

Kandidatenschlüssel bestimmen:

$$\kappa_1 = \{A, G\}, \kappa_2 = \{F, G\}, \kappa_3 = \{C, G\}, \kappa_4 = \{E, G\}$$

Kandidatenschlüssel der Relationenzerlegung hinzufügen:

Da keiner der Kandidatenschlüssel in der Zerlegung enthalten ist,  
wähle ein beliebiges  $\kappa_i$  und erstelle  $\mathcal{R}_\kappa$ , z. B.  $\mathcal{R}_\kappa = \{\underline{A}, G\}$ .

Redundanzen entfernen:

$$\mathcal{R}_4 \text{ verwerfen, da } \mathcal{R}_4 \subseteq \mathcal{R}_1$$

Ergebnis:

$$\{\underline{A}, C, D, E\}, \{A, \underline{F}\}, \{B, \underline{E}, F\}, \{\underline{A}, G\}$$