



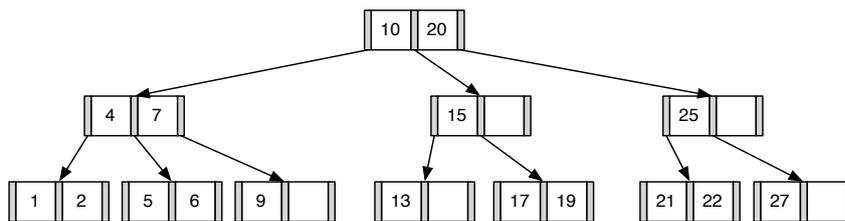
## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS17/18

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

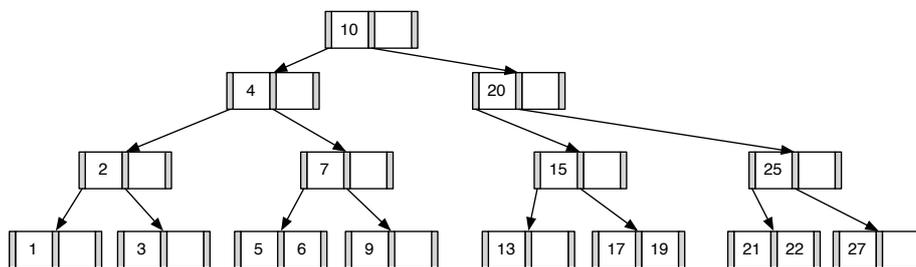
<http://www-db.in.tum.de/teaching/ws1718/grundlagen/>

Blatt Nr. 10

### Hausaufgabe 1

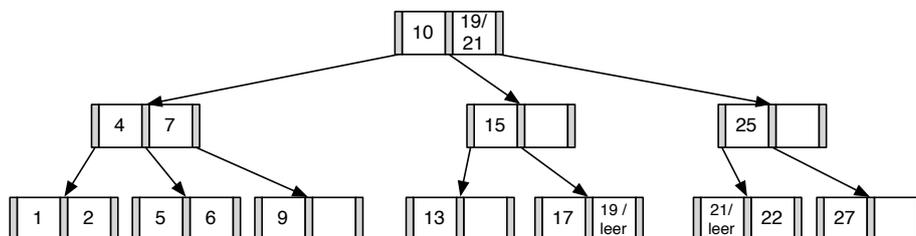


1. Fügen Sie die 3 in den gezeigten B-Baum ein. Zeichnen Sie das Endergebnis. Zeichnen Sie jeweils den kompletten Baum oder machen Sie **deutlich**, falls Teile des Baumes unverändert bleiben. Verwenden Sie den aus der Vorlesung bekannten Algorithmus. Das Ergebnis sieht wie folgt aus:



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 2

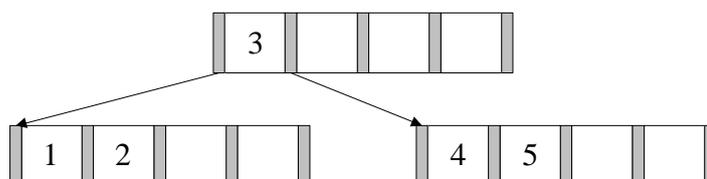
Fügen Sie in einen anfänglich leeren B-Baum mit  $k = 2$  die Zahlen eins bis zwanzig in aufsteigender Reihenfolge ein. Was fällt Ihnen dabei auf?

**Lösung:**

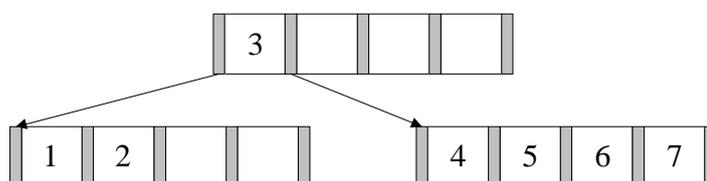
Nachdem man die Zahlen 1 bis 4 eingefügt hat, liegt folgender B-Baum vor:



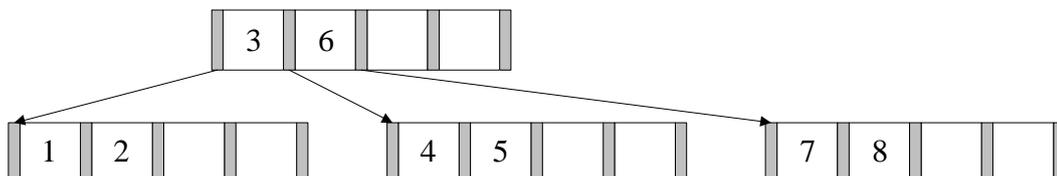
Beim Einfügen von 5 wird der Knoten gespalten und man erhält eine neue Wurzel.



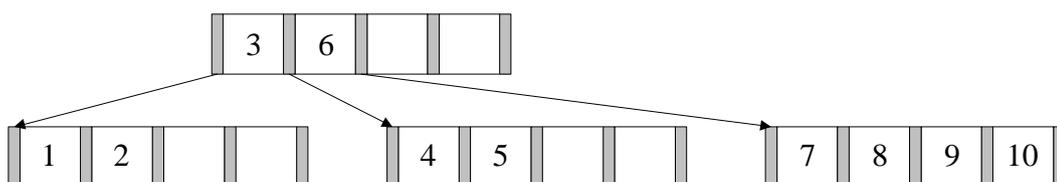
Die nächsten beiden Zahlen lassen sich wieder ohne Probleme einfügen.



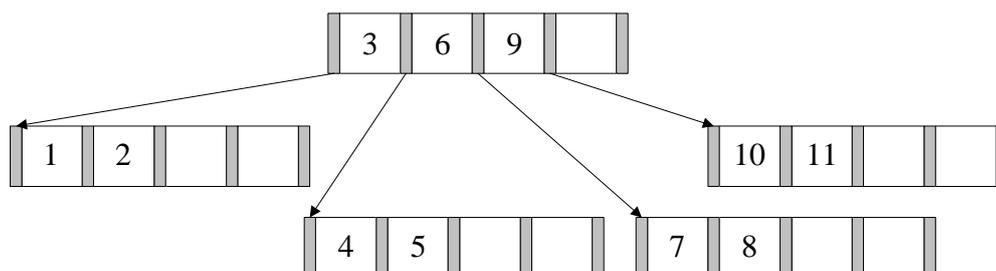
Beim Einfügen der 8 kommt es erneut zum Überlauf. Die 6 wandert in die Wurzel.



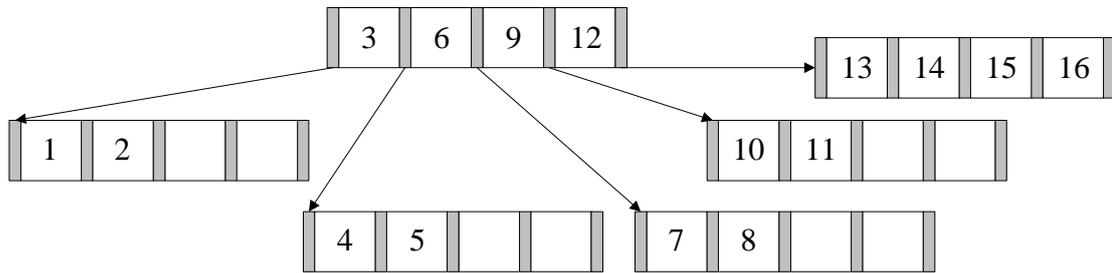
9 und 10 lassen sich wieder ohne Probleme einfügen. Bei 11 kommt es zum Überlauf.



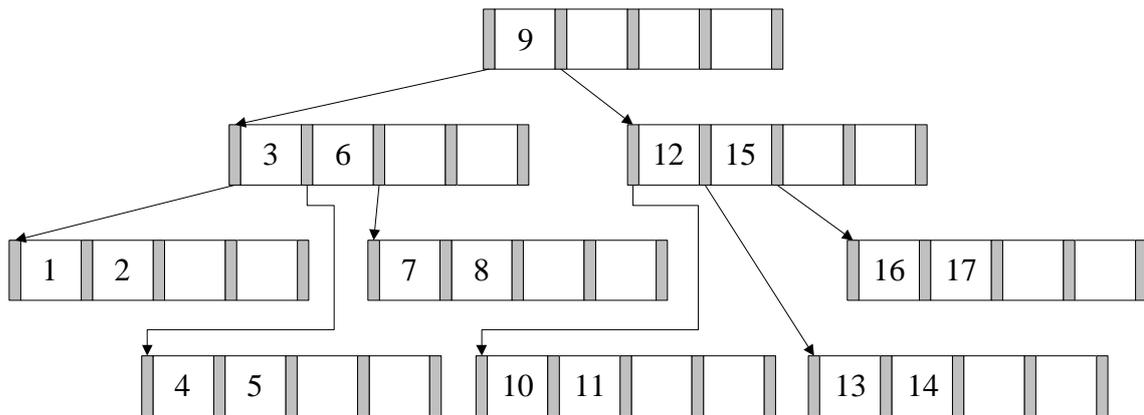
Nach dem Aufspalten erhält man dann:



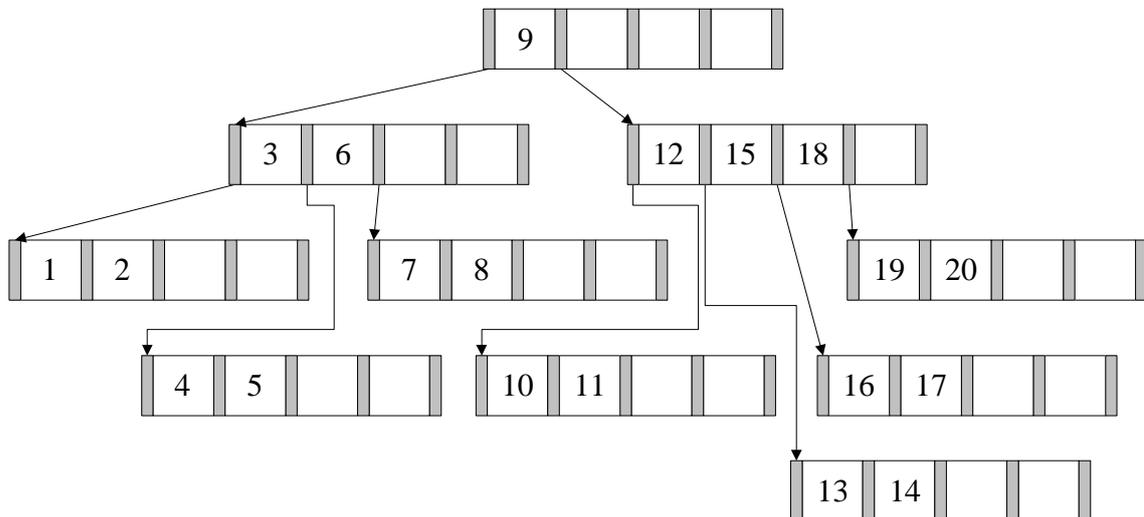
Es werden nun die nächsten Zahlen bis 16 analog eingefügt.



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 3

Fügen Sie nacheinander die folgenden Einträge in eine anfangs leere erweiterbare Hashtabelle, welche 2 Einträge pro Bucket aufnehmen kann, ein. Es soll effizient nach der **KundenNr** gesucht werden können.

KundenNr	Name
10	Müller
25	Meier
30	Schmidt
18	Krause
40	Schulz
45	Kaufmann

Die Hashwerte<sup>1</sup> für die jeweiligen Attributwerte entnehmen Sie den beiden folgenden Tabellen:

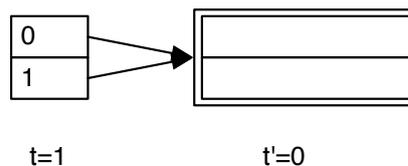
Name	Hash(Name)	KundenNr	Hash(KundenNr)
Müller	101100	10	010100
Meier	001011	25	100110
Schmidt	110000	30	011110
Krause	111111	18	010010
Schulz	000011	40	000101
Kaufmann	100000	45	101101

### Lösung:

Werte mit binärer KundenNr sowie invers binärer Kundennummer, die für das Einfügen in den Hash genutzt wird:

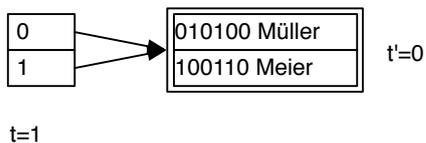
KundenNr	Name	Binär	Umgekehrt Binär
10	Müller	001010	010100
25	Meier	011001	100110
30	Schmidt	011110	011110
18	Krause	010010	010010
40	Schulz	101000	000101
45	Kaufmann	101101	101101

Zunächst eine leere erweiterbare Hashtabelle:

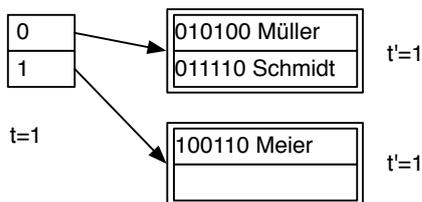


<sup>1</sup><https://de.wikipedia.org/wiki/Hashfunktion>

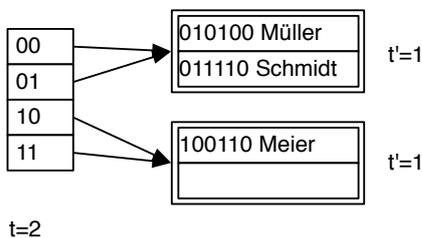
Wir fügen nun die ersten zwei Einträge ein, wonach die Hashtabelle wie folgt aussieht:



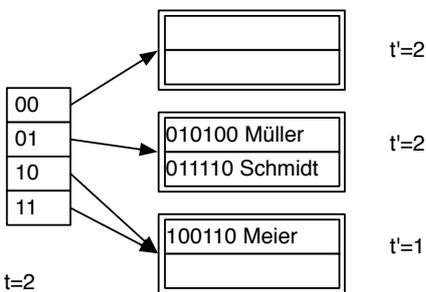
Der nächste Eintrag führt zu einem Überlauf. Da  $t' < t$  können wir den Bucket teilen, dies führt zur folgenden Hashtabelle:



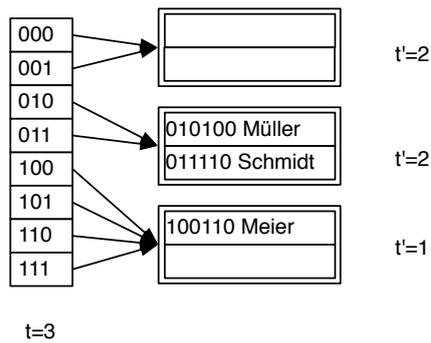
Das Einfügen von Krause führt erneut zu einem Überlauf, der Bucket kann aber nicht direkt geteilt werden, da  $t' = t$  gilt. Das Verzeichnis wird verdoppelt:



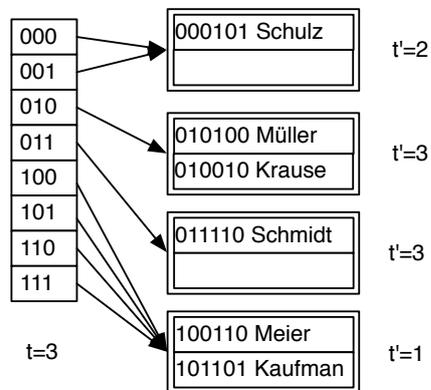
Nun kann der Bucket geteilt werden:



Das Einfügen ist leider immer noch nicht möglich und wieder hilt  $t' = t$ , weswegen das Verzeichnis erneut verdoppelt werden muss:



Nun kann der Bucket geteilt und alle Einträge eingefügt werden:



### Quiz 1

Geben Sie eine Permutation der Zahlen 1 bis 24 an, so dass beim Einfügen dieser Zahlenfolge in einen (anfänglich 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