



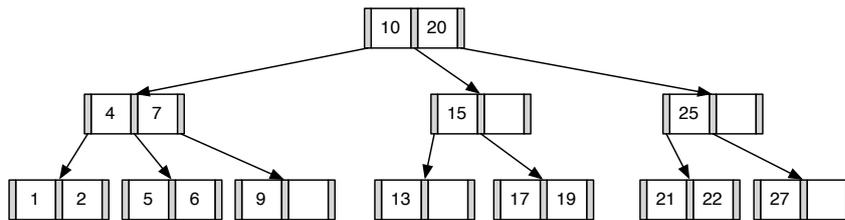
Übung zur Vorlesung *Grundlagen: Datenbanken* im WS19/20

Christoph Anneser, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1920/grundlagen/>

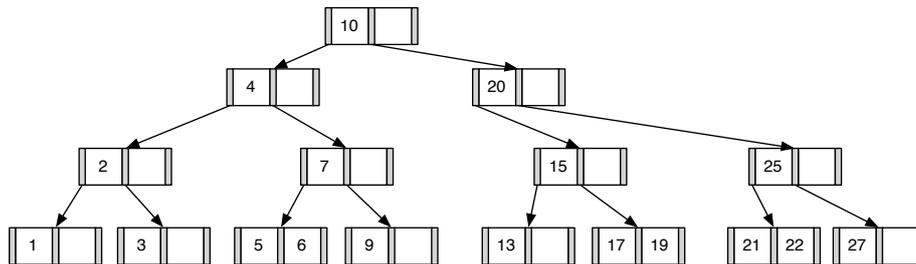
Blatt Nr. 10

Hausaufgabe 1



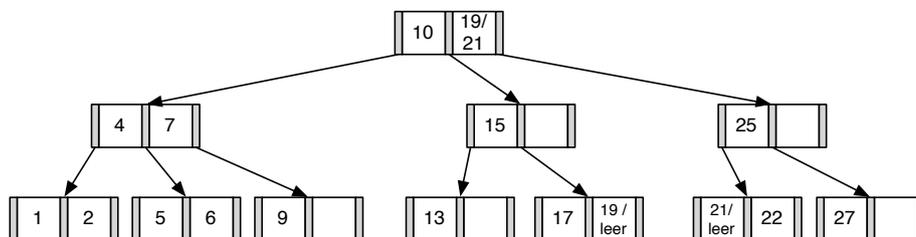
1. Fügen Sie die 3 in den gezeigten B-Baum ein. Zeichnen Sie das Endergebnis. Zeichnen Sie jeweils den kompletten Baum oder machen Sie **deutlich**, falls Teile des Baumes unverändert bleiben. Verwenden Sie den aus der Vorlesung bekannten Algorithmus.

Lösung: 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.

Lösung: Das Ergebnis sieht wie folgt aus:

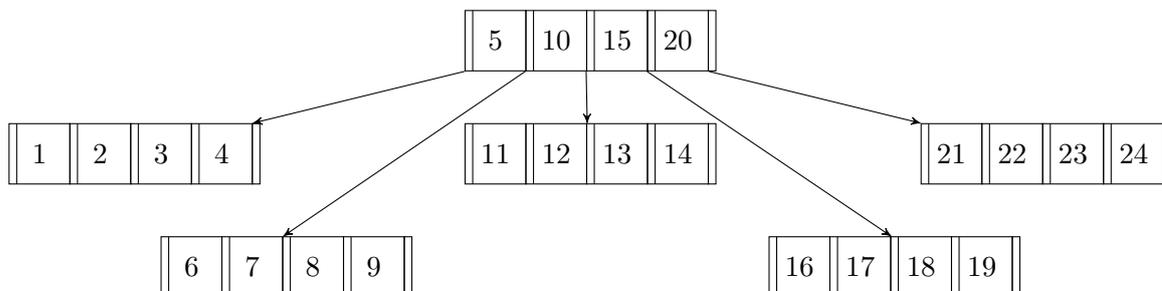


Hausaufgabe 2

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.

Lösung:

Damit der Baum eine minimale Höhe erreicht, muss der Füllgrad aller Knoten möglichst hoch sein. Bei $k = 2$ passen vier Elemente in jeden Knoten. Bei 24 einzufügenden Zahlen bedeutet das also, dass ein minimaler Baum insgesamt aus $\frac{24}{4} = 6$ Knoten besteht, die alle komplett gefüllt sind. Aufgrund der Invarianzen eines B-Baums muss dies ein Baum der Höhe 2 sein. Der einzig mögliche minimale Baum ist der folgende:



Damit die Zahlen 5, 10, 15 und 20 in die Wurzel geschrieben werden, muss die Einfügereihenfolge so gewählt werden, dass beim Teilen eines Knotens wegen Überlauf genau diese vier Zahlen als mittlere Elemente verwendet werden. Außerdem muss darauf geachtet werden, dass die Knoten nach dem Aufteilen wieder voll gefüllt werden.

Beispielsweise kann beim Einfügen mit den Zahlen 1, 2, 5, 6 und 7 in dieser Reihenfolge angefangen werden, wodurch die 5 dann in die Wurzel geschoben wird. Um die 10 in die Wurzel zu schreiben, können dann die Zahlen 10, 11, 12 eingefügt werden. Für die 15 dementsprechend die Zahlen 15, 16, 17 sowie für die 20 die Zahlen 20, 21, 22. Die übrigen Zahlen können dann in beliebiger Reihenfolge eingefügt werden.

Eine mögliche Permutation der Einfügereihenfolge, um einen Baum minimaler Höhe zu erhalten ist also: 1, 2, 5, 6, 7, 10, 11, 12, 15, 16, 17, 20, 21, 22, 3, 4, 8, 9, 13, 14, 18, 19, 23, 24.

Hausaufgabe 3

Es sollen die folgenden Informationen aller Menschen auf der Erde (ca. 10 Milliarden) in einer erweiterbaren Hashtabelle verwaltet werden: Steuernummer, Name, Land, Stadt, PLZ, Straße und Hausnummer (insgesamt ca. 100 B). Dabei ist die Steuernummer eindeutig und 10 B lang und wird in der Hashtabelle als Suchschlüssel verwendet.

Der Suchschlüssel wird direkt in den Buckets der Hashtabelle gespeichert. Die restlichen Daten eines Tupels hingegen werden nicht direkt in den Buckets sondern auf separaten Seiten gespeichert. Auf diese wird mithilfe von TIDs der Größe 8 B verwiesen. Um auf Buckets zu verweisen, verwendet die Hashtabelle Page IDs, die ebenfalls 8 B groß sind.

Gehen Sie bei der Berechnung davon aus, dass jedes Bucket der Hashtabelle auf einer Speicherseite der Größe 4 KiB gespeichert wird und ein Bucket durchschnittlich zu 75% gefüllt ist.

Wie viel Speicherplatz verbraucht das Verzeichnis für diese Hashtabelle mindestens?

(1 KiB = 2^{10} B)

Lösung:

Jeder Eintrag eines Buckets enthält die Steuernummer und die TID der restlichen Daten. In einem Bucket sind also durchschnittlich $n = \frac{4 \text{ KiB}}{10 \text{ B} + 8 \text{ B}} \cdot 0.75 (\approx 170)$ Einträge gespeichert.

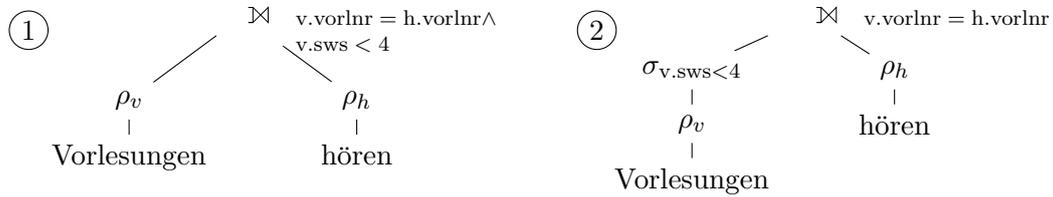
Für den gesamten Datensatz werden also $p = \frac{10^{10}}{n}$ ($\approx 58 \cdot 10^6$) Seiten benötigt. Um p Seiten zu referenzieren, werden $\lceil \log_2 p \rceil = 26$ Bits benötigt. Um 26 Bits referenzieren zu können, enthält das Verzeichnis mindestens 2^{26} Verweise. Jeder Verweis auf ein Bucket wird als Page ID der Größe 8 B gespeichert. Also hat das Verzeichnis eine Größe von mindestens $2^{26} \cdot 8 \text{ B} = 512 \text{ MiB}$.

Beachten Sie, dass die Größe eines Datensatzes irrelevant für diese Berechnung ist, da TIDs verwendet werden. Nur die Länge des Suchschlüssels ist ausschlaggebend, da diese die Anzahl der Tupel, die pro Bucket gespeichert werden können, beeinflusst.

Hausaufgabe 4

Klausuraufgabe aus dem WiSe 2018/19:

Gegeben seien die beiden folgenden Algebraausdrücke in Operatorbaumdarstellung:



Sind die beiden Algebraausdrücke äquivalent? Begründen Sie!

Lösung:

Die Ausdrücke sind nicht äquivalent, da die beiden Bäume unterschiedliche Ergebnisse liefern.

Die Selektion filtert Vorlesungen aus, während der left outer join Tupel der linken Seite, die die Join-Bedingung nicht erfüllen (also z.B. von der Selektion ausgefiltert worden wären) mit null-Werten für die rechte Seite in die Ergebnismenge aufnimmt.

Mit der Beispielausprägung enthält das Ergebnis des linken Baumes z.B. die Vorlesung Ethik, der rechte Baum aber nicht.