

Prof. Dr. Agn'es Voisard, Nicolas Lehmann

Datenbanksysteme, SoSe 2017

Übungsblatt 6

Tutor: Nicolas Lehmann
Tutorium 10

Boyan Hristov, Julian Habib

18. Juni 2017

Link zum Git Repository: <https://github.com/BoyanH/Freie-Universitaet-Berlin/tree/master/Datenbanksysteme/Solutions/homework6>

1. Aufgabe

a) z.z.: R_1 ist in 1NF \Leftrightarrow alle Attribute sind atomar.

$R_1(A, B, C, D) \subseteq A \times B \times C \times D \Rightarrow A, B, C, D$ sind atomar.

Alle Attributen in R_1 haben atomäre Domäne, sind also keine Relationen. Z.B. alle Einträge haben für den Attribut A die Werte a_1, a_2, a_3 und keine Werte wie z.B. (a_1, a_2) .

□

b) z.z.: R_1 ist nicht in 2NF

R_1 ist in 2NF $\Leftrightarrow R_1$ ist in 1NF und $\forall(X \rightarrow A) \in F_+ | A \notin X : X \not\subset$ Superschlüssel $\forall A \subseteq$ Schlüssel.

Beweis durch Gegenbeispiel:

$$\begin{aligned}
 FD(R_1) = \{ & \\
 & A \rightarrow B \\
 & A \rightarrow C \\
 & D \rightarrow C \\
 & BD \rightarrow A \\
 \} &
 \end{aligned}$$

\Rightarrow AD und BD sind Superschlüssel und auch beide Kandidatschlüsseln.

Primäre Attributen sind A, B und D.

Da C kein Primärattribut ist, von A abhängig ist und da A eine Untermenge eines Schlüssels ist (AD), ist R_1 nicht in 2NF.

□

- c) z.z.: R_2 ist in 3NF $\Leftrightarrow \forall FD(X \rightarrow A) : X$ ist Superschlüssel von R oder A ist Primärattribut von R
- Direkter Beweis:

$$FD(R_2) = \{$$

$$E \rightarrow F$$

$$E \rightarrow G$$

$$FG \rightarrow E$$

}

$\Rightarrow E$ und FG sind Superschlüsselelemente, Primärattributen sind E , F und G .

In den ersten zwei funktionalen Abhängigkeiten ist die linke Seite ein Superschlüssel, in der 3. Abhängigkeit ist die rechte Seite ein Primärattribut. $\Rightarrow R_2$ ist in 3NF.

□

- d) z.z.: R_2 ist in BCNF $\Leftrightarrow \forall X \rightarrow A \in F_+ : X$ ist Superschlüssel.

Direkter Beweis:

$$FD^+(R_2) = \{$$

$$E \rightarrow F$$

$$E \rightarrow G$$

$$E \rightarrow FG$$

$$FG \rightarrow E$$

}

Da wir schon aus c) kennen, dass E und FG Superschlüsselelemente sind und da diese alle möglichen linken Seiten von einer funktionalen Abhängigkeit sind, ist R_2 in Boyce-Codd Normalform (BCNF).

□

2. Aufgabe

- a) Mit dem Algorithmus aus der Vorlesung für "Well-behaved 3NF Decomposition"

$$FD(R_3) = \{$$

$$H \rightarrow JK$$

$$I \rightarrow HJ$$

$$K \rightarrow L$$

}

1. For each FD $(X \rightarrow A)$ in F create a relation with schema (XA) .

$$(HJK), (IHJ), (KL)$$

2. If none of the keys appears in one of the schemas of 1 then add a relation with schema Y, with Y a key.

\Rightarrow fertig, da H ein Schlüssel ist und in (HJK) vorhanden ist.

3. If for relations created in 1. there exists a relation R1 whose schema is included in the schema of another relation, then remove R1.

Alles ist in Ordnung, die Attributen von keinem Schema sind eine Untermenge von den Attributen eines anderen Schemas.

4. Replace relations (X A1), ..., (X Ak) with a single relation (X A1 ... Ak).

$$(HJK) \wedge (IHJ) \Rightarrow (HJIK)$$

Gute Zerlegung: $R_{31}(H, J, I, K)$ und $R_{32}(K, L)$

b)

$$\begin{aligned} R_{31} \cap R_{32} &\equiv K \\ R_{31} - R_{32} &\equiv HJI \end{aligned}$$

$$R_{32} - R_{31} \equiv L$$

$$R_{31} \cap R_{32} \rightarrow KL \text{ (wegen } K \rightarrow L\text{)} \rightarrow L \equiv R_{31} - R_{32}$$

\Rightarrow Unsere Zerlegung ist verlustlos

c)

$$\begin{aligned} FD(R_{31}) &= \{H \rightarrow JK, I \rightarrow HJ\} \\ FD(R_{32}) &= \{K \rightarrow L\} \\ FD(R_{31}) \cup FD(R_{32}) &\equiv \{H \rightarrow JK, I \rightarrow HJ, K \rightarrow L\} \equiv FD(R_3) \\ \Rightarrow \text{Unsere Zerlegung ist abhängigkeitserhaltend} \end{aligned}$$

- d) Da wir dieses Algorithmus benutzt haben, ist jede davon entstandene Relation in 3NF. Wegen b), c) und d) ist die Zerlegung gut. Für R_{32} sind H und I Schlüssel und an alle funktionalen Abhängigkeiten auf der linken Seite vorhanden, deswegen ist R_{32} in 3NF. R_{32} hat nur eine funktionale Abhängigkeit und nur 2 Attributen, ist deswegen trivialerweise in 3NF.

3. Aufgabe

- a) Aus jedem Item kann man durch eine Hash-Funktion ein Hash erzeugen durch mathematische Umformungen aus allen Suchschlüssel. Mehrere Items werden in dem selben Bucket gespeichert (nacheinandere folgende Blöcke im Speicher) wenn die Hashfunktion das gleiche Hash aus ihren Suchschlüssel erzeugt. Danach, beim Suchen von Lösch / Einfüge / Lese Position müssen alle Einträge in dem entsprechenden Bucket nach einander geprüft werden. Deswegen erzeugen gute Hashfunktionen randomisierte Hashes, damit es ungefähr genau so viele Items pro Bucket gibt.

- b) In einem Dense-Index gibt es Indizes für alle Suchschlüsselwerte. Z.b. wenn das Attribut A ein Primärindex ist, gibt es für alle mögliche Werte von A ein Index, die Einträge mit dem gleichen Wert von A stehen nach einander im Speicher.

Der Sparse-Index hat nur für gewählte Werte von den Suchschlüssel Indizes. Dabei müssen aber immer die Einträge im Speicher nach dem Primärindex sortiert werden. Damit sucht man nach dem Index des alphabetisch größten kleineren Wert und muss dann alle weitere Einträge durchsuchen, um den Item zu finden. Dabei haben wir viel speicher für den Index gespart, die Suchzeiten haben sich aber vergrößert. Ein gutes Balance zwischen den beiden ist zu finden.

- c) Ein Multilevel-Index ist ein Index, wo das unterste Niveau ein Dense-Index ist und darüber noch n Stufen von Sparse-Index existieren. So kann man viel schneller suchen ($\log_2(n)$), da es so viele Schritte vom Wurzel zu den Blättern in einem balancierten Baum mit n Blättern gibt). Die Blätter eines B^+ Baums sind die äußeren Indizes und die restlichen Knoten die inneren Indizes. Diese Struktur darf man aber nur dann verwenden, wenn die Suchschlüssel sortierbar sind, da man sonst kein B^+ oder überhaupt kein Suchbaum konstruieren kann.
 - d) "Fixed Length Records"
Sei 3 die Länge eines Records

#record	Film Name	Actor's Name	Actor's Address
0	Fremde Gewässer	Thomas Depp	Paris
1	Fremde Gewässer	Linda Cruz	Los Angeles
2	Furious Sixteen	Martin Diesel	Detroit
3	Furious Sixteen	Grünther Walker	New York

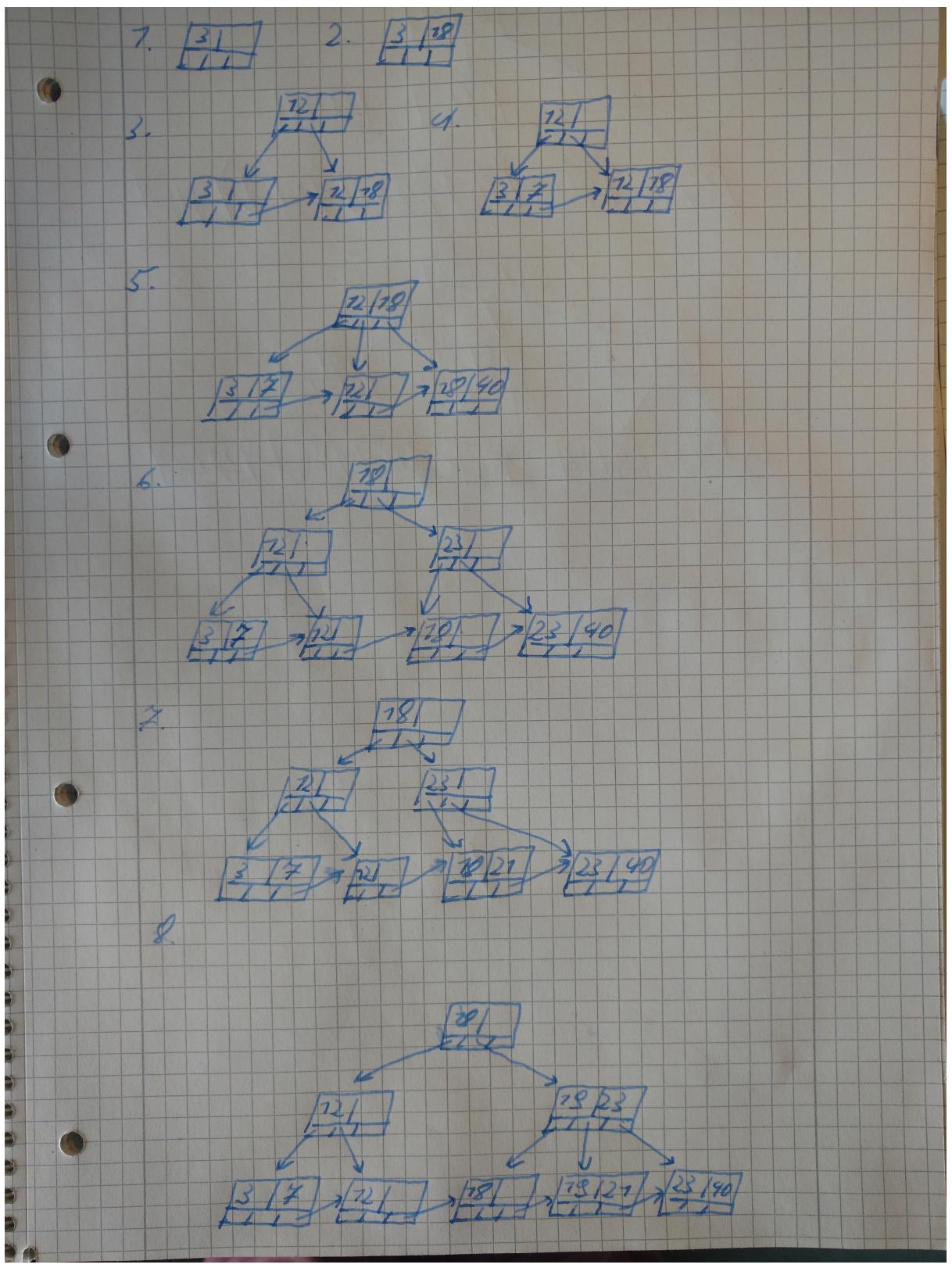
”Variable Length Records”

#record	Film Name	Actor's Name	Actor's Address	Actors's Name
0	Fremde Gewässer	Thomas Depp	Paris	Linda Cruz	Los Angeles	⊥
1	Furious Sixteen	Martin Diesel	Detroid	Grünther Walker	New York	⊥

Mit Hilfe von Pointer

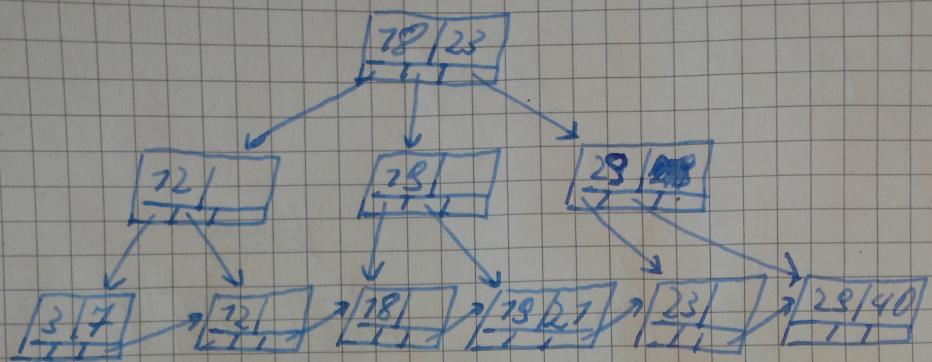


Aufgabe 3.

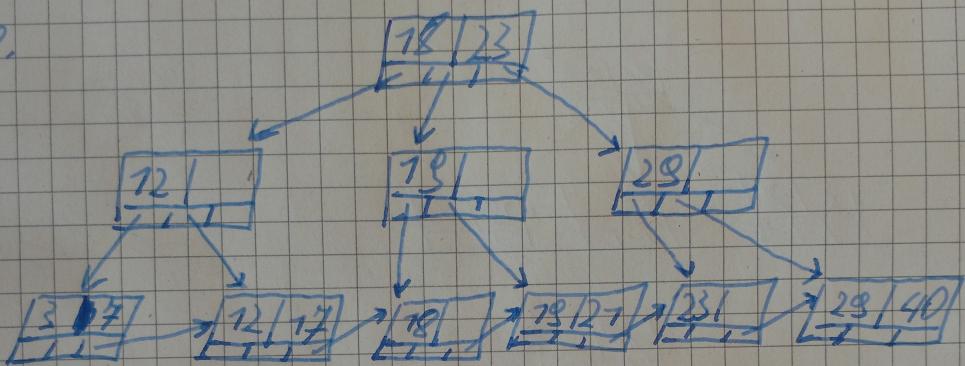


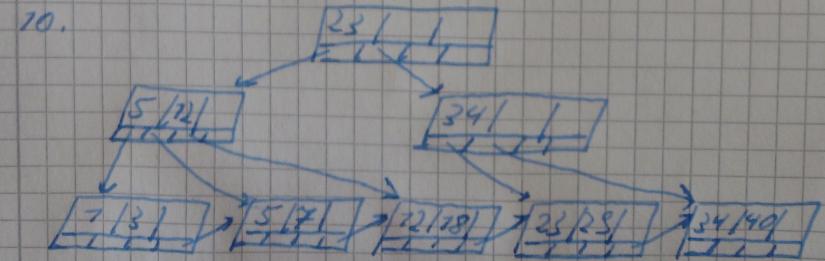
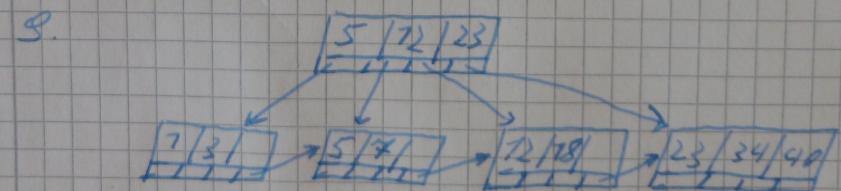
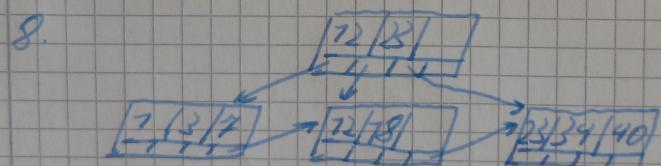
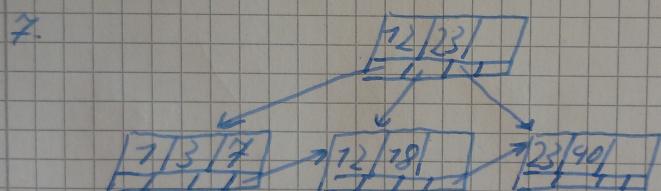
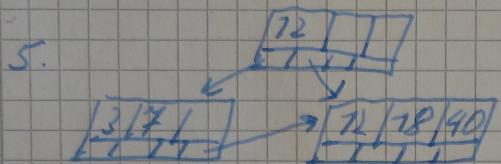
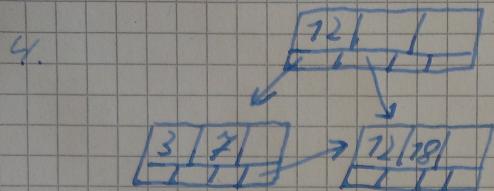
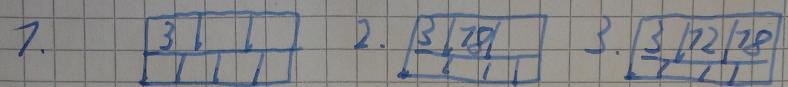
a)

9.

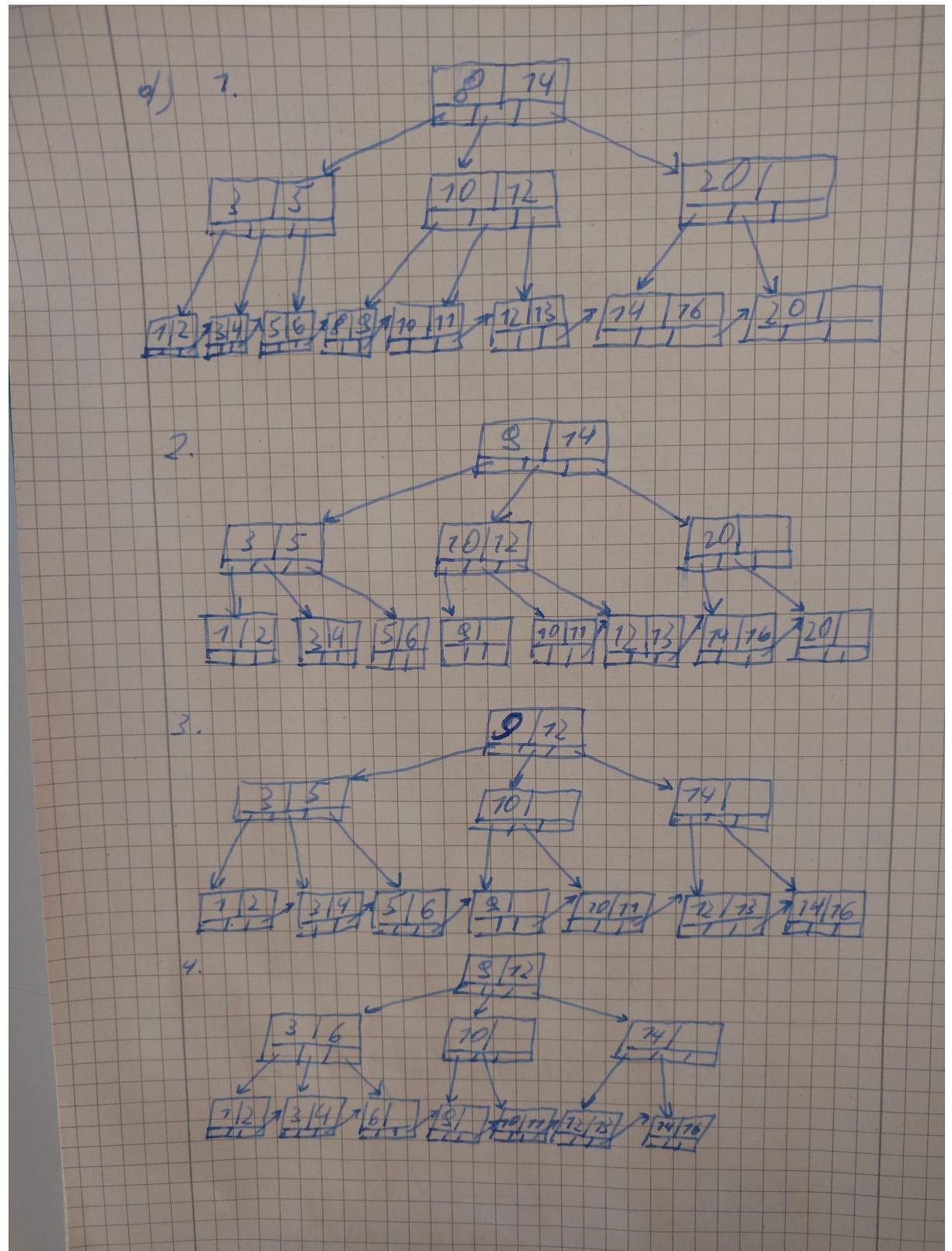


10.





b)



5.

