

## Aufgabenblatt 2 Indexstrukturen

- Abgabetermin: **Mittwoch, 01.12.10 (23:59 Uhr)**
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabe:
  - per E-Mail an `dbs2-2010@hpi.uni-potsdam.de` mit Subject  
Abgabe DBS II: Aufgabenblatt <n> Namen
  - ausschließlich pdf-Dateien
  - eine Datei pro Aufgabe mit folgendem Dateinamen:  
`blatt<aufgabenblattNr>aufgabe<aufgabenNr><Nachnamen>.pdf`  
Bitte **keine Leerzeichen, Unterstriche, Umlaute, Sonderzeichen, ...** im Dateinamen!
  - **jedes Blatt beschriftet mit Namen und Matrikelnummern**
  - Wir korrigieren die Abgaben aufgabenweise. Das beschriebene Verfahren vereinfacht uns die Arbeit erheblich!

### Aufgabe 1: Indexstrukturen auf sequentiellen Dateien

Sei  $R(\underline{A}, B, C)$  eine Relation.  $R$  umfasst 12 Tupel, die nach dem Attribut  $A$  sortiert in einer Datei gespeichert sind (sequentielle Datei). Innerhalb eines Blocks können zwei Tupel von  $R$  gespeichert werden.

Die nachfolgende Abbildung zeigt die Blockverteilung der Tupel von  $R$  in der sequentiellen Datei, wobei für jedes Tupel nur der Wert des Attributs  $A$  angegeben wurde.

(22, *, *)
(24, *, *)
(26, *, *)
(28, *, *)
(30, *, *)
(32, *, *)
(34, *, *)
(36, *, *)
(38, *, *)
(40, *, *)
(42, *, *)
(44, *, *)

- a) Indiziere die Datei mit einem zweistufigen Primärindex  $\mathcal{I}$ , wobei in jeder Stufe ein dünnbesetzter Index zu verwenden ist. Innerhalb eines Blocks können vier Schlüssel-Zeiger-Paare gespeichert werden. 3 P

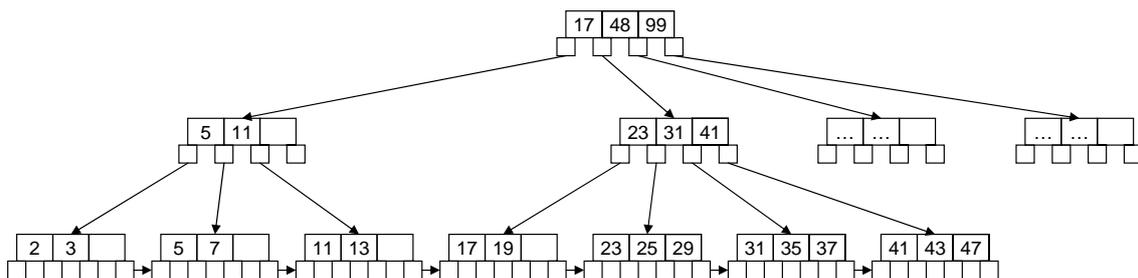
- b) Notiere zuerst in der Abbildung aus Teilaufgabe a) neben jedem Block der Indexstruktur und der sequentiellen Datei einen eindeutigen Bezeichner (z. B. eine Zahl). Betrachte nun die nachfolgenden SQL-Anfragen und gebe jeweils die eindeutigen Bezeichner der einzulesenden Blöcke an, die für die Anfrageausführung *unter Verwendung des zweistufigen Primärindex  $\mathcal{I}$*  eingelesen werden müssen.

*Hinweis:* Beachte, dass  $A$  Primärschlüssel der Relation  $R$  ist. Zudem befinden sich keine Blöcke von  $R$  und keine Blöcke von  $\mathcal{I}$  im Hauptspeicher. 7 P

- 1) SELECT \* FROM R WHERE A = 34
- 2) SELECT \* FROM R WHERE A > 36
- 3) SELECT COUNT(\*) FROM R WHERE A = 38
- 4) SELECT COUNT(\*) FROM R WHERE A = 40
- 5) SELECT MIN(A) FROM R
- 6) SELECT MAX(A) FROM R
- 7) SELECT AVG(A) FROM R

## Aufgabe 2: Einfügen und Löschen in $B^+$ -Bäumen

Gegeben sei der folgende  $B^+$ -Baum:



- a) Füge die Schlüssel 1, 4, 18 und 20 (in dieser Reihenfolge) ein. Gib den  $B^+$ -Baum nach dem Einfügen der 4 und der 20 an. Beschreibe kurz für jeden Schlüssel die notwendigen Schritte zum Einfügen. 10 P
- b) Lösche die Schlüssel 1 bis 5 und 7 (in dieser Reihenfolge). Gib den  $B^+$ -Baum jeweils nach dem Löschen der 4 und 7 an. Beschreibe kurz für jeden Schlüssel die notwendigen Schritte zum Löschen. 10 P

## Aufgabe 3: Bulk Loading

Gegeben sei eine Relation, die 20.000 Tupel umfasst. Die Relation wurde mittels eines  $B^+$ -Baums indiziert. Zur Erstellung des Index wurde die in der Vorlesung vorgestellte *Bulk Loading*-Methode verwendet, wobei die Knoten jeder Ebene jeweils so voll wie möglich gefüllt wurden.

Für die Speicherung eines Suchschlüssels werden 40 Byte und für die Speicherung eines Zeigers 10 Byte benötigt. Die Größe eines Blocks beträgt 1000 Byte.

- a) Wie groß ist der Parameter  $n$ ? 2 P
- b) Wieviele Ebenen hat der erzeugte  $B^+$ -Baum? 3 P
- c) Gib für jede Ebene die Anzahl der Knoten an. 3 P
- d) Wieviele Ebenen hätte der  $B^+$ -Baum, wenn durch Komprimierung für den Suchschlüssel durchschnittlich nur noch 10 Byte benötigt würden? 2 P