
Aufgabe 1 - Festplattenparameter.**1 Punkt**

Betrachten Sie die folgende Festplatte:

- Sektorgröße $B = 2^9$ Bytes,
- Interblock Gap $G = 0$ Bytes,
- Spuren pro Scheibenseite $T = 300$,
- Anzahl der beidseitig beschriebenen Scheiben $D = 2^3$,
- mittlerer Spurwechsel $sp = \frac{1}{480}$ s,
- Zugriffszeit pro Sektor für einen (lesenden) Random Access $tz = \frac{1.961}{384.000}$ s,
- Drehzahl $dz = 10^4$ rpm.

Berechnen Sie die **Sektoren pro Spur (S)** der Festplatte.

Lösung: _____

Aufgabe 2 - Slotted Page.**1 Punkt**

Gegeben ist die folgende Slotted Page:

- Größe: 2^{14} Bytes
- Header: **dense** (dicht gepackt)
- Adressierungstyp: **Byte-Adressierung**

Berechnen Sie die **Größen der Felder im Header** (a , f , g_i und p_i) und weiters **wieviele Datensätze der Größe 2^3 Bytes auf der Slotted Page gespeichert werden können.**

Lösung: _____

Aufgabe 3 - Mehrstufiger Index.

1 Punkt

Auf der folgenden Tabelle soll ein **zweistufiger Index** auf dem Attribut **Affiliation** konstruiert werden. D.h. der Index hat 2 flache Index-Stufen, wobei die **innere Stufe dense** und die **äußere Stufe sparse** sein soll. **Pro Indexblock** können **3 Einträge** gespeichert werden.

Zeichnen Sie den zweistufigen Index für die gegebene Tabelle.

Name	Affiliation	Area
Arya	CMU	DBS
Cersei	PLUS	DBS
Daario	TUM	HPC
George	Google	CC
Jon	SU	OS
Melisandre	CMU	DBS
Robb	SU	SWS
Shae	PLUS	GM
Tyrion	CMU	SWS
Ygritte	SU	HPC

Name:

Matrikelnummer:

5/??

Aufgabe 4 - B^+ -Baum-Konstruktion.

1 Punkt

Gegeben sind die Blätter eines B^+ -Baumes ($m = 5$). Konstruieren Sie die **darüberliegenden Ebenen** (d.h. die inneren Knoten) so, dass ein **gültiger B^+ Baum mit minimaler Höhe** entsteht.



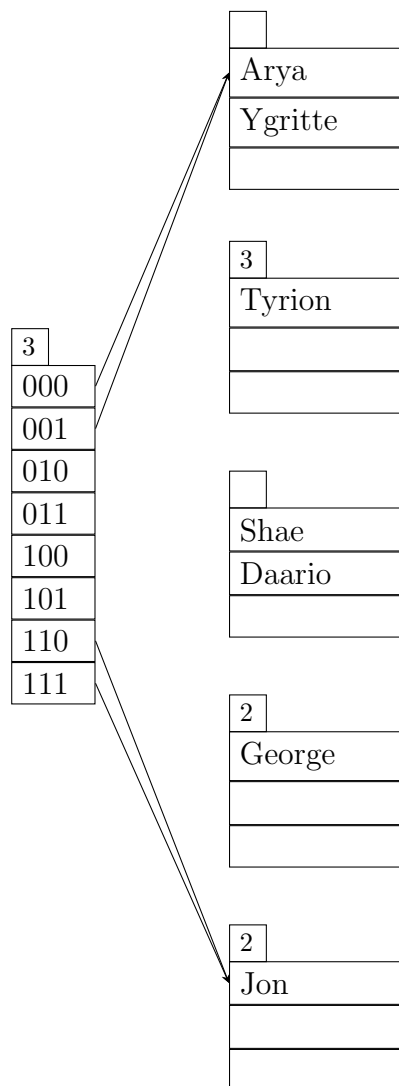
Aufgabe 5 - Erweiterbares Hashing.

1 Punkt

Die Hashfunktion h liefert die in der Tabelle angegebenen Binärwerte. **Ergänzen Sie die fehlenden Werte/Komponenten (Werte, lokale Tiefen und Pointer).**

Hash-Tabelle:

Name (x)	$h(x)$ (binär)
Arya	0010
Cersei	0100
Daario	0110
George	1000
Jon	1100
Melisandre	1110
Robb	1011
Shae	0111
Tyrion	0101
Ygritte	0001



Name:

Matrikelnummer:

7/??

Aufgabe 6 - *Externes Merge-Sort*.

1 Punkt

Führen Sie **externes Merge-Sort** auf der folgenden Relation $R[A]$ aus.
Jeder **Block fasst 2 Tupel**. Die Größe des **Puffer** beträgt $M = 4$ **Blöcke**.

72
108
48
42
60
102
90
12
84
66
0
114
96
30
54
78
18
36
24
120
6

Aufgabe 7 - Join-Algorithmen.**1 Punkt**

Gegeben seien **zwei Relationen** mit folgenden Eigenschaften:

$R[A, B, C]$:

- $|R| = 10^7$ Tupel, gespeichert auf $b_R = 20 \cdot 10^3$ Datenblöcken
- **dense** B⁺-Baum-Index auf Attribut A , $m = 2^8$
- **sparse** B⁺-Baum-Index auf Attribut B , $m = 2^7$
- Die B⁺-Bäume besitzen **maximale Höhe**.

$S[B, D, F]$:

- $|S| = 5 \cdot 10^6$ Tupel, gespeichert auf $b_S = 4 \cdot 10^3$ Datenblöcken
- **flacher dense** Index auf Attribut B mit $5 \cdot 10^4$ Indexblöcken
- **flacher sparse** Index auf Attribut D mit 40 Indexblöcken

Es soll ein **Natural Join** $R \bowtie S$ unter Anwendung des **Indexed-Nested-Loop-Join-Algorithmus** durchgeführt werden.

Geben Sie hierfür die effizienteste Join-Reihenfolge ($R \bowtie S$ oder $S \bowtie R$) sowie die zugehörigen **Kosten (Blockzugriffe)** an. Ein Knotenzugriff im B⁺-Baum entspricht einem Blockzugriff. Duplikate können für diese Aufgabe vernachlässigt werden.

Lösung: _____

Aufgabe 8 - Effiziente Anfragebearbeitung.**1 Punkt**

Gegeben sei eine Relation $R[A, B]$ mit folgenden Eigenschaften:

- $|R| = 10^7$ Tupel, gespeichert auf $b_R = 2.000$ Datenblöcken.
- Die Attributwerte von A und B sind positive ganze Zahlen.
- Duplikate sind erlaubt.
- Attribut A ist gleichverteilt im Bereich $[1; 10^7]$.
- Attribut B ist gleichverteilt im Bereich $[1; 10^5]$.
- Es existiert ein **sparse B⁺-Baum-Index auf Attribut A** mit Knotengrad $m = 2.048$. Der B⁺-Baum hat **minimale Höhe**. Ein Knotenzugriff im B⁺-Baum entspricht einem Blockzugriff.
- Es existiert ein **flacher dense Index auf Attribut B**, wobei **pro Indexblock 10³ Einträge** gespeichert werden können.

Die folgende Anfrage soll ausgewertet werden:

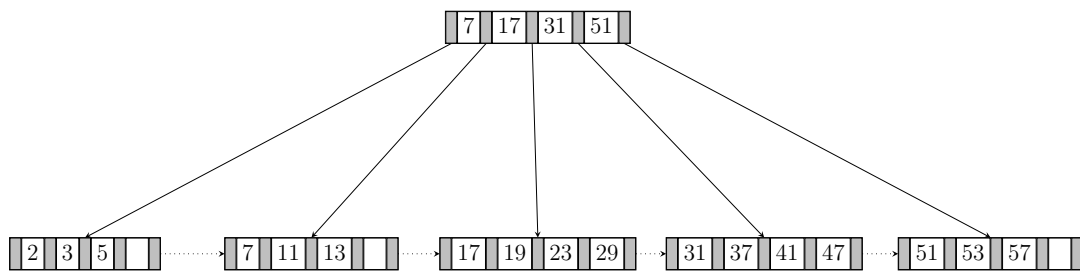
$$\sigma_{B < 10^3 \wedge A > 10^6} (R)$$

Geben Sie die **effizienteste Strategie und deren Kosten (Blockzugriffe)** an, um die gegebene Anfrage auszuwerten.

Aufgabe 9 - B^+ -Baum.

1 Punkt

Gegeben ist der unten abgebildete B^+ -Baum mit $m = 5$. Fügen Sie den Wert 43 gemäß des in der Vorlesung besprochenen Algorithmus in diesen B^+ -Baum ein.



Aufgabe 10 - Anfrageoptimierung, Join-Kardinalitäten.**1 Punkt**

Gegeben seien die folgenden 3 Relationen $R[A, B, C]$, $S[A, B, V]$ und $T[A, B, X]$:

- $|R| = 10^3$ Tupel, $V(R, A) = 250$, $V(R, B) = 10$, $V(R, C) = 1.000$.
- $|S| = 10^4$ Tupel, $V(S, A) = 200$, $V(S, B) = 250$, $V(S, V) = 2.000$.
- $|T| = 10^5$ Tupel, $V(T, A) = 2.500$, $V(T, B) = 1.000$, $V(T, X) = 3.000$.

Weiters sei die folgende SQL-Anfrage gegeben:

```
SELECT  R.A, T.X
FROM    R, S, T
WHERE   R.A = S.A
        AND R.B = S.B
        AND S.A = T.A
        AND S.B = T.B
```

1. Zeichnen Sie die **algebraische Normalform als Operatorbaum** für die gegebene SQL-Anfrage. **(0.5 Punkte)**
2. Wenden Sie **heuristische Optimierung** an, um den **Operatorbaum zu optimieren**. Im resultierenden Operatorbaum soll auch die **Join-Reihenfolge optimal** sein, d.h. es soll zuerst der Join mit dem kleinsten Zwischenergebnis durchgeführt werden. **(0.5 Punkte)**