
 Aufgabe 1 - *Indexstrukturen*.

1 Punkt

Auf der Tabelle $R[\textit{Musician}, \textit{Stage}]$ soll ein **Hash-Index** konstruiert werden. Als Schlüssel wird das Attribut *Musician* verwendet. Der Hashwert ist der **numerische Wert** des **1. Buchstaben** des Attributwerts. Es können **3 Tupel pro Bucket** gespeichert werden. Bucket Overflows werden durch **Overflow Chaining** aufgelöst, wobei ein Zeiger auf ein Overflow Bucket einen Eintrag im Bucket benötigt. **Illustrieren** Sie den **Hash-Index**.

1. Buchstabe	D	E	M	N	P	S	T
numerischer Wert	1	2	3	4	5	6	7

Musician	Stage
Noisia	Mothership
Madface	Temple
Merikan	Shredder
Pythius	Portal
Signal	Shredder
Mefjus	Portal
Emperor	Portal
Mizo	Shredder
Turno	Temple
Dimension	Mothership

Aufgabe 2 - *Slotted Page*.

1 Punkt

Gegeben sei eine Slotted Page mit folgenden Eigenschaften:

- Größe: $2^{13} = 8192$ Bytes,
- Adressierungstyp: **Byte-Adressierung** (es kann jedes Byte adressiert werden)

In dieser Slotted Page werden **3 Tupel** O , P , Q gespeichert:

- d_1 : $|O| = 64$ Bytes
- d_2 : $|P| = 128$ Bytes
- d_3 : $|Q| = 256$ Bytes

Ergänzen Sie die Slotted Page um die **fehlenden Werte/Adressen**, wobei p_i und g_i sich auf den jeweiligen Datensatz d_i beziehen. (**0.125 Punkte pro Wert/Adresse**)

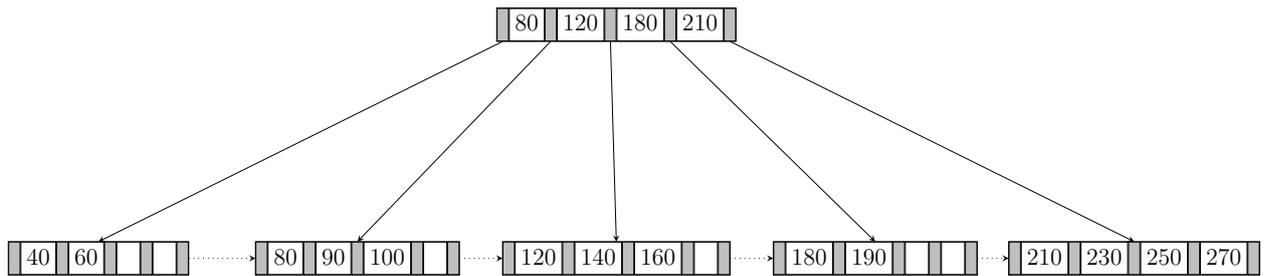
a f g_3 p_3 g_2 p_2 g_1 p_1 \dots d_1 d_2 d_3

										O	P	Q
--	--	--	--	--	--	--	--	--	--	-----	-----	-----

Aufgabe 3 - B⁺-Baum Löschen.

1 Punkt

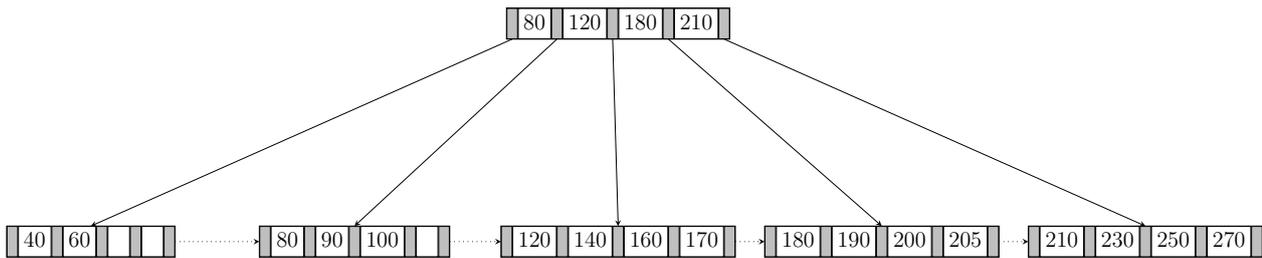
Gegeben ist ein B⁺-Baum mit $m = 5$. Zeichnen Sie den B⁺-Baum, der nach Löschen von 180 entsteht.



Aufgabe 4 - B^+ -Baum-Einfügen.

1 Punkt

Gegeben ist ein B^+ -Baum mit $m = 5$. Zeichnen Sie den B^+ -Baum, der nach dem Einfügen von **260** entsteht.



Aufgabe 5 - Erweiterbares Hashing.

1 Punkt

Gegeben ist der folgende Hashcontainer, in dem 10 Tupel gespeichert sind. Ein Bucket im Hashcontainer kann bis zu 3 Tupel speichern. Die Hashfunktion $h(x)$ liefert die in der Tabelle angegebenen Binärwerte. **Ergänzen Sie die fehlenden Zeiger, sodass ein gültiger Hashcontainer entsteht. (0.125 Punkte pro korrekt verbundenem Bucket)**

x	$h(x)$
<i>Noisia</i>	0000
<i>Madface</i>	0010
<i>Merikan</i>	0001
<i>Pythius</i>	1110
<i>Signal</i>	0011
<i>Mefjus</i>	1100
<i>Emperor</i>	1000
<i>Mizo</i>	0101
<i>Turno</i>	1111
<i>Dimension</i>	0011

4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

3
Noisia
Merikan

4
Madface

4
Dimension
Signal

2
Mizo

2
Emperor

3
Mefjus

4
Pythius

4
Turno

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 **3 Blöcke**.

99
35
10
45
95
20
25
5
30
90
40
0
15
50
80
60
55
75
65
70
85

Aufgabe 7 - Effiziente Anfragebearbeitung.**1 Punkt**

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

- $|R| = 4 \cdot 10^9$ Tupel,
- Pro Datenblock werden $8 \cdot 10^3$ Tupel gespeichert,
- Attribut A hat ganzzahlige Werte gleichverteilt im Bereich $[1; 4 \cdot 10^9]$,
- Attribut B hat ganzzahlige Werte gleichverteilt im Bereich $[1; 4 \cdot 10^6]$,
- Attribut C hat ganzzahlige Werte gleichverteilt im Bereich $[1; 4 \cdot 10^5]$,
- Duplikate werden mittels Tuple Identifier (TID) aufgelöst,
- Folgende Indizes existieren:
 - flacher sparse Index auf Attribut A mit 12.500 Einträgen pro Indexblock,
 - dense B^+ -Baum-Index auf Attribut B , $m = 2.048$, minimale Höhe,
 - dense B^+ -Baum-Index auf Attribut C , $m = 1.024$, minimale Höhe.

Es soll folgende Anfrage beantwortet werden:

$$\sigma_{C=8 \cdot 10^3 \wedge A \geq 4 \cdot 10^3} (R)$$

Geben Sie die **Strategie (0.5 Punkte)** an und berechnen Sie die **Anzahl der Blockzugriffe (0.5 Punkte)** um die Anfrage **möglichst effizient** zu beantworten (1 Knotenzugriff im B^+ -Baum entspricht 1 Blockzugriff).

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

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

- $|R| = 8 \cdot 10^9$ Tupel,
- Pro Datenblock werden $16 \cdot 10^3$ Tupel gespeichert,
- Attribut A hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^6]$,
- Attribut B hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^9]$,
- Attribut C hat ganzzahlige Werte gleichverteilt im Bereich $[1; 8 \cdot 10^3]$,
- Duplikate werden mittels Tuple Identifier (TID) aufgelöst,
- Folgende Indizes existieren:
 - flacher sparse Index auf Attribut A mit 12.500 Einträgen pro Indexblock,
 - dense B^+ -Baum-Index auf Attribut B , $m = 4.096$, minimale Höhe,

Es soll folgende Anfrage beantwortet werden:

$$\sigma_{B < 8 \cdot 10^9}(R)$$

Geben Sie die **Strategie (0.5 Punkte)** an und berechnen Sie die **Anzahl der Blockzugriffe (0.5 Punkte)** um die Anfrage **möglichst effizient** zu beantworten (1 Knotenzugriff im B^+ -Baum entspricht 1 Blockzugriff).

Aufgabe 9 - Join-Algorithmen.**1 Punkt**

Welcher Join Algorithmus (**Hash Join, Block Nested Loop Join (Zick-Zack Modus)**) generiert die minimalen Kosten für das folgende Szenario? Geben Sie in der Lösung die **Algorithmen und die dazugehörigen Kosten** an.

Berechnen Sie einen Natural Join zwischen zwei Relationen $R[A, B]$ und $S[B, C]$, wobei $|R| = 4.000$ Tupel und $|S| = 1.000$ Tupel. Die Relationen sind auf $b_R = 600$ bzw. $b_S = 1.000$ hintereinander liegenden Blöcken gespeichert. Der Buffer hat Platz für $M = 21$ Blöcke und $k = 4$. Es existiert ein sparse B⁺-Baum Index auf $R.B$, wobei jeder Knoten im B⁺-Baum 20 Schlüssel speichern kann.

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

Gegeben sind die folgende Relationen:

- $|R[A, B, C]| = 100.000$, $V(R, A) = 150$, $V(R, B) = 50$, $V(R, C) = 50$
- $|S[C, D, E]| = 5.000$, $V(S, C) = 20$, $V(S, D) = 50$, $V(S, E) = 100$
- $|T[A, B]| = 10.500$, $V(T, A) = 80$, $V(T, B) = 70$

Die Werten in den Tupeln sind gleichverteilt und unabhängig.

1. Formulieren Sie die folgende Anfrage so um, dass Selektionen und Projektionen so weit wie möglich nach unten geschoben werden (sodass der kleinstmögliche Join durchgeführt wird). Geben Sie entweder den umformulierten relationalen Ausdruck oder den entsprechenden Operatorbaum an.

$$\pi_A(\pi_{B,C,D}(\sigma_{A=a \wedge D=d}(R \bowtie S)) \bowtie T)$$

2. Schätzen Sie die Kardinalität des Ergebnisses der Joins $R \bowtie S$ und $R \bowtie T$ ab.