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

Gegeben ist die folgende Slotted Page:

- Größe: 2^{16} 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 2^4 Bytes auf der Slotted Page gespeichert werden können.**

Lösung: _____

Aufgabe 2 - *Free List*.

1 Punkt

Gegeben ist eine Tabelle $R[Name]$ mit 10 Tupel, die als Datei mit fixer Datensatzlänge organisiert ist. **Gelöschte Einträge** werden über eine **Free List** verwaltet. Es werden die folgenden Einträge gelöscht (in dieser Reihenfolge):

5 (Falcon), 7 (Iron Man), 2 (Captain America).

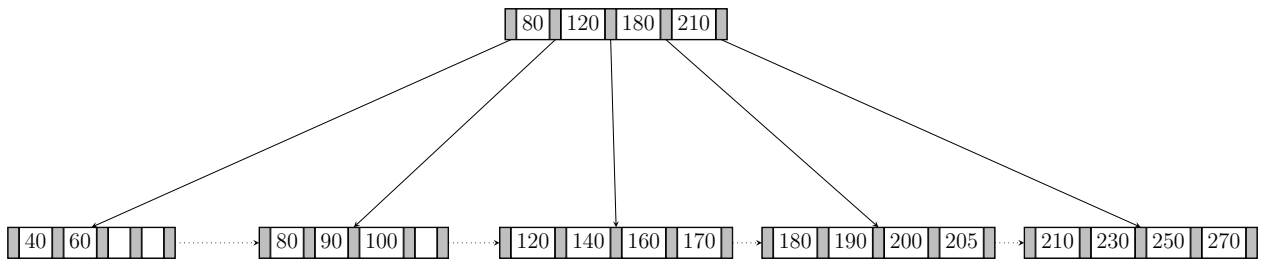
header	
record 0	Ant-Man
record 1	Black Panther
record 2	Captain America
record 3	Deadpool
record 4	Doctor Strange
record 5	Falcon
record 6	Hulk
record 7	Iron Man
record 8	Spider-Man
record 9	Thor

Geben Sie die **resultierende Datei** an und **zeichnen** Sie die **Free List** ein (inklusive Start und Ende).

header	
record 0	
record 1	
record 2	
record 3	
record 4	
record 5	
record 6	
record 7	
record 8	
record 9	

Aufgabe 3 - B⁺-Baum-Konstruktion.**1 Punkt**

Gegeben ist ein B⁺-Baum mit $m = 5$. Zeichnen Sie den B⁺-Baum, der nach dem Löschen von **60** entsteht.



Aufgabe 4 - Effiziente Anfragebearbeitung.**1 Punkt**

Gegeben ist die Relation $R[A, B]$. Auf $R.A$ existiert ein sparse B⁺-Baum Index und auf $R.B$ existiert ein dense Hash-Index. Die Werte für Attribut B sind eindeutig. Was ist die **effizienteste Strategie** um Anfragen von folgendem Typ zu beantworten?

$$\sigma_{A < a \vee B = b}(R)$$

Geben Sie **alle notwendigen Schritte** an.

Aufgabe 5 - Effiziente Anfragebearbeitung.**1 Punkt**

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

- $|R| = 10 \cdot 10^{12}$ Tupel,
- Pro Datenblock werden $25 \cdot 10^3$ Tupel gespeichert,
- Attribut A hat ganzzahlige Werte gleichverteilt im Bereich $[1; 20 \cdot 10^6]$,
- Attribut B hat ganzzahlige Werte gleichverteilt im Bereich $[1; 20 \cdot 10^9]$,
- Attribut C hat ganzzahlige Werte gleichverteilt im Bereich $[1; 20 \cdot 10^3]$,
- Duplikate werden mittels Tuple Identifier (TID) aufgelöst.
- Folgende Indizes existieren:
 - einstufiger sequentieller sparse Index auf Attribut A mit $5 \cdot 10^6$ Einträgen pro Indexblock,
 - dichte B^+ -Baum-Index auf Attribut B , $m = 1.024$, minimale Höhe.

Es soll folgende Anfrage beantwortet werden:

$$\sigma_{A=7 \cdot 10^3 \vee B=7 \cdot 10^3}(R)$$

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 6 - *Externes Merge-Sort*.1 Punkt

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

100
35
10
45
95
20
25
5
30
90
40
0
15
50
80
60
55
75
65
70

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

Gegeben seien zwei Relationen mit folgenden Eigenschaften:

- $R[A]$ mit Werten 1, 2, 3, 4, 5, 6, 7, 8, 9,
- $S[A]$ mit Werten 2, 5, 6, 8, 9, 11, 13, 15, 16, 18, 19, 20,
- die Größe des Puffer beträgt $M = 4$ Blöcke,
- jeder Block fasst 1 Tupel.

Für die Durchführung eines Hash Joins soll $h(x) = x \bmod 3$ als Hashfunktion verwendet werden. Erklären Sie, ob und warum diese Hashfunktion für die Durchführung eines Hash Joins auf die Attribute $R[A]$ und $S[A]$ unter den gegebenen Umständen verwendet werden kann.

Aufgabe 8 - Join-Algorithmen.**1 Punkt**

Die Relation $R[A, B, C]$ hat $n_R = 3000$ Tupel auf $b_R = 500$ Blöcken, die Relation $S[B, D]$ hat $n_S = 2000$ Tupel auf $b_S = 500$ Blöcken. Es soll ein natürlicher Join mittels Block Nested Loop Joins durchgeführt werden.

Berechnen Sie die Anzahl an Blockzugriffen basierend auf unterschiedlichen Parametern:

- Es steht ein Puffer der Größe $M = 2$ zur Verfügung.
- Es steht ein Puffer der Größe $M = 501$ zur Verfügung.
- Es wird der Zick-Zack-Modus mit $M = 501$ und $k = 200$ verwendet.

Aufgabe 9 - Anfrageoptimierung.**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)**

Aufgabe 10 - Join Kardinalität.**1 Punkt**

Gegeben sind folgende Relationen:

- $|R[A, B, C]| = 1000$, $V(R, A) = 100$, $V(R, B) = 200$, $V(R, C) = 300$
- $|S[A, D, E]| = 4000$, $V(S, A) = 50$, $V(S, D) = 200$, $V(S, E) = 300$
- $|T[C, D, F]| = 2000$, $V(T, C) = 100$, $V(T, D) = 200$, $V(T, F) = 600$

Die Werte in den Tupeln sind gleichverteilt und unabhängig. Schätzen Sie die Kardinalität des Ergebnisses der folgenden Anfrage ab ($\sigma_{A=100 \vee A=200}(R) \neq \emptyset$).

$$(\sigma_{A=100 \vee A=200}(R)) \bowtie S \bowtie T$$