
 Aufgabe 1 - *Slotted Page*.

1 Punkt

Gegeben sei eine Slotted Page mit folgenden Eigenschaften:

- Größe: $2^{12} = 4096$ Bytes
- Adressierungstyp: **Word-Adressierung** (es kann nur jedes 2. Byte adressiert werden)

In dieser Slotted Page werden die Tupel U , C , K gespeichert:

- d_1 : $|U| = 827$ Bytes
- d_2 : $|C| = 722$ Bytes
- d_3 : $|K| = 242$ Bytes

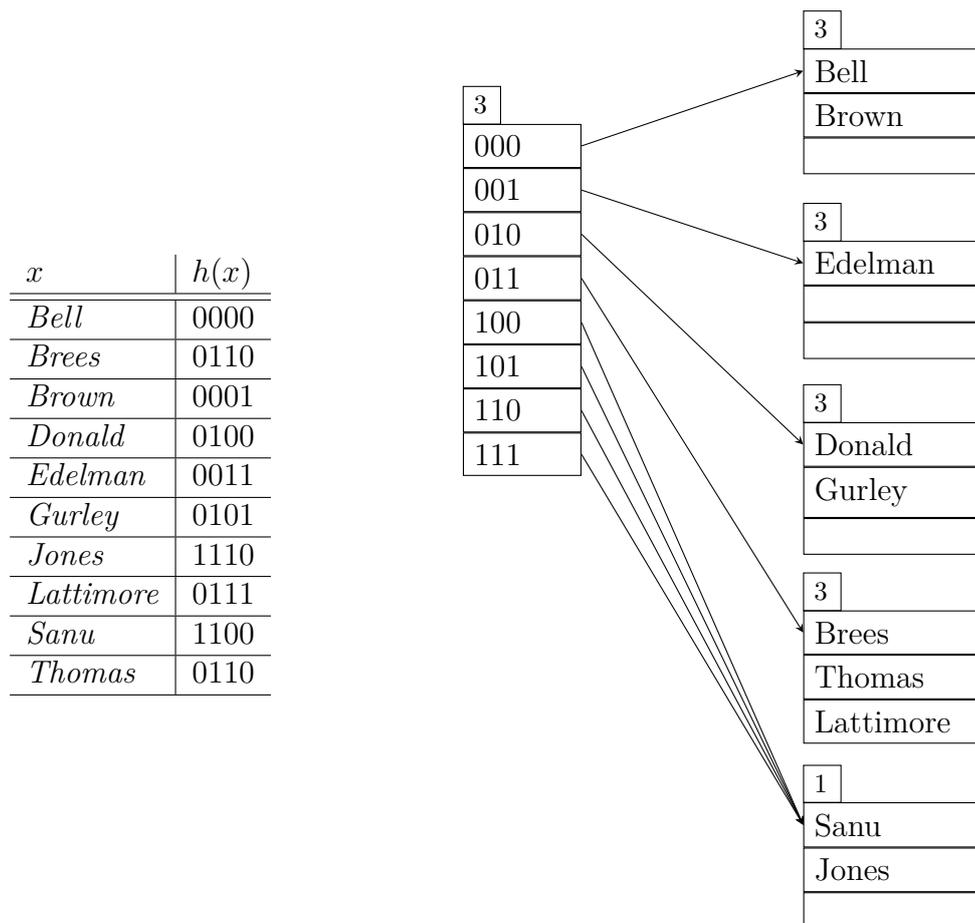
Ergänzen Sie die Slotted Page um die **fehlenden Werte/Adressen** (numerische Werte erwartet, Pfeile reichen nicht aus), wobei p_i und g_i sich auf den jeweiligen Datensatz d_i beziehen.

a	f	g_1	p_1	g_2	p_2	g_3	p_3	...	d_1	d_2	d_3
									U	C	K

Aufgabe 2 - Erweiterbares Hashing.

1 Punkt

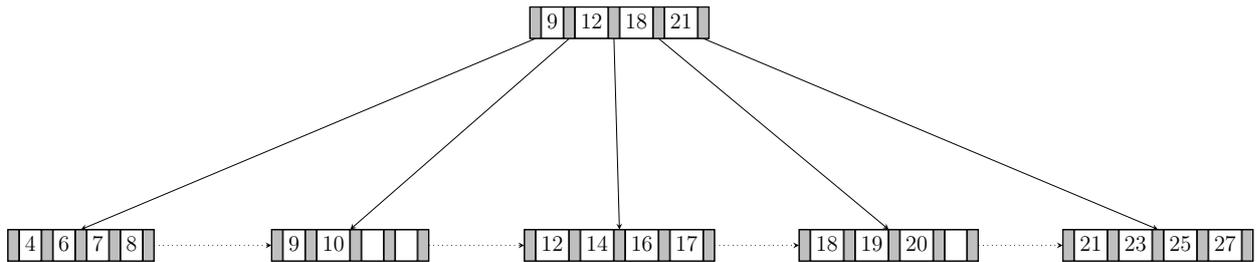
Die Hashfunktion $h(x)$ liefert die in der Tabelle angegebenen Binärwerte. Es soll das Tupel **Donald** aus dem gegebenen Hashcontainer gelöscht werden. Ein Bucket im Hashcontainer kann bis zu 3 Tupel speichern. Das Verzeichnis soll so klein wie möglich gehalten werden. **Illustrieren Sie den resultierenden Hashcontainer.**



Aufgabe 3 - B⁺-Baum Löschen.

1 Punkt

Gegeben ist ein B⁺-Baum mit $m = 5$. Zeichnen Sie den B⁺-Baum, der nach dem Löschen von **10** entsteht.



Aufgabe 4 - Indexstrukturen.**1 Punkt**

Zeichnen Sie für die folgende Tabelle einen **2-stufigen Clustered Index** auf dem Attribut **Player**. **Beide Indexstufen** sollen **sparse** sein. Ein **Datenblock** kann **2 Tupel** speichern. In einen **Indexblock** können **5 Einträge** gespeichert werden.

Player	Team
Bell	Steelers
Brees	Saints
Brown	Steelers
Donald	Rams
Edelman	Patriots
Fournette	Jaguars
Gurley	Rams
Houston	Chiefs
Jones	Falcons
Kuechly	Panthers
Lattimore	Saints
Leonard	Colts
Mack	Bears
Michel	Colts
Sanu	Falcons
Thomas	Saints

Aufgabe 5 - Bitmap Index Scan.**1 Punkt**

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

- $|R| = 10^6$ Tupel gespeichert auf $b_R = 10^3$ Blöcken,
- Non-clustering B⁺-Baum-Index auf Attribut B , $m = 1024$, maximale Höhe (d.h. die Knoten sind minimal befüllt),
- Duplikate werden mittels Tuple Identifier (TID) aufgelöst,
- Attribut B hat insgesamt 10^3 mal den Wert 2500, verteilt auf 100 Datenblöcke.

Es soll folgende Anfrage beantwortet werden:

$$\sigma_{B=2500}(R)$$

Geben Sie die Anzahl an **Datenblockzugriffen (worst case)**

- a. **ohne Anwendung** eines Bitmap Index Scans (**0.5 Punkte**),
- b. **unter Anwendung** eines Bitmap Index Scans (**0.5 Punkte**)

an.

Aufgabe 6 - Join-Algorithmen.**1 Punkt**

Gegeben seien **zwei Relationen** R and S 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 und
- jeder Block fasst 1 Tupel.

Zeichnen Sie die Partitionen für einen hash join für beide Relationen. Erklären Sie Schritt für Schritt wie die Partitionen entstehen.

Aufgabe 7 - 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,
 - Dense 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 8 - Schätzung der Join-Kardinalität.

1 Punkt

Gegeben seien 3 Relationen $R[A, B, C]$, $S[B, C, D, F]$, $T[A, D, E, F]$ mit folgenden Eigenschaften:

- $|R| = 2500$ Tupel, $V(R, A) = 50$, $V(R, B) = 10$, $V(R, C) = 20$
- $|S| = 500$ Tupel, $V(S, B) = 50$, $V(S, C) = 40$, $V(S, D) = 10$, $V(S, F) = 100$
- $|T| = 4000$ Tupel, $V(T, A) = 25$, $V(T, D) = 50$, $V(T, E) = 200$, $V(T, F) = 20$

Die Werte in den Tupeln sind gleichverteilt und unabhängig. Schätzen Sie die Kardinalität der folgenden Abfrage ab.

$$R \bowtie S \bowtie T$$

Aufgabe 9

1 Punkt

Ist der folgende Anfrageplan **conflict serializable**? Falls dies **zutrifft**, geben Sie einen äquivalenten seriellen Anfrageplan an. Falls es **nicht zutrifft**, begründen Sie ihre Antwort.

T1:	T2:	T3:	T4:
-----			read(B)
	read(A)		-----
	write(A)		-----
read(C)			-----
			read(A)
		read(C)	-----
		read(B)	-----
			read(C)
		write(A)	-----
		write(B)	-----
write(B)			-----

Aufgabe 101 Punkt

Ist der folgende Anfrageplan kompatibel mit **two phase locking (2PL)**? Falls **ja**, tragen Sie alle **lock** und **unlock** Instruktionen in den Anfrageplan ein. Falls **nein**, erklären Sie warum 2PL verletzt wird.

T1: T2: T3:

 read(B)

 read(C)

 write(A)

read(A)

 write(C)