

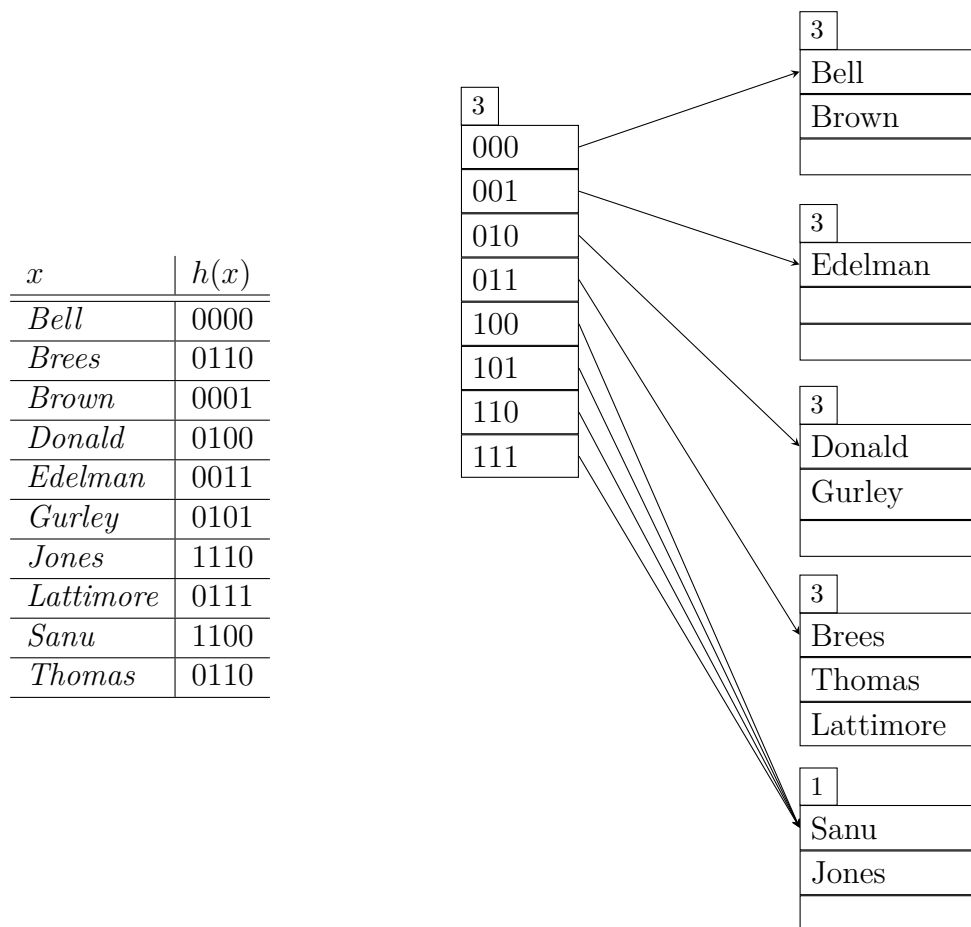




## Aufgabe 2 - Erweiterbares Hashing.

1 Punkt

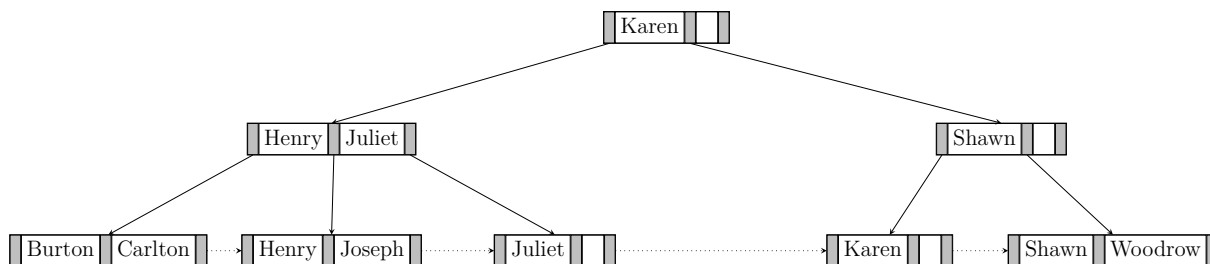
Die Hashfunktion  $h(x)$  liefert die in der Tabelle angegebenen Binärwerte. Es soll das Tupel **Donald** aus dem gegebenen Hashcontainer gelöscht werden. Ein Bucket im Hashcontainer kann bis zu 3 Tupel speichern. Das Verzeichnis soll so klein wie möglich gehalten werden. **Illustrieren Sie den resultierenden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



Aufgabe 3 - B<sup>+</sup>-Baum Einfügen.

1 Punkt

Gegeben ist ein B<sup>+</sup>-Baum mit  $m = 3$ . Zeichnen Sie den B<sup>+</sup>-Baum, der nach dem Einfügen von **Isaac** entsteht.



---

**Aufgabe 4 - Indexstrukturen.****1 Punkt**

---

**Zeichnen** Sie für die folgende Tabelle einen **3-stufigen Sekundärindex** auf dem Attribut **Stadt**. Die **innere Indexstufe** soll **dense** und die **äußeren beiden Indexstufen** sollen **sparse** sein. In einen **Indexblock** können **3 Einträge** gespeichert werden.

Stadt	KFZ
Rom	I
London	GBM
Prag	CZ
Kiew	UA
Berlin	D
Athen	GR
Krakau	PL
Oslo	N
Dublin	IRL
Wien	A

---

**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<sup>+</sup>-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

---

Welcher Join Algorithmus (**Merge Join**, **Index Nested Loop Join**) generiert die minimalen Kosten für das folgende Szenario? Geben Sie in der Lösung die **Algorithmen** und die **dazugehörigen Kosten** an.

Berechnen Sie einen **natürlichen Join** zwischen zwei Relationen  $R[A, B]$  und  $S[A, C]$  mit  $|R| = 1000$  Tupel und  $|S| = 12000$  Tupel. Die Relationen sind auf  $b_R = 250$  bzw.  $b_S = 2000$  hintereinander liegenden Blöcken gespeichert. Es existiert ein **sparse B<sup>+</sup>-Baum** Index auf  $S.A$ , wobei jeder Knoten im B<sup>+</sup>-Baum 20 Schlüssel speichern kann. Auf  $R.A$  existiert kein Index, falls Sie eine Sortierung benötigen, nehmen Sie an, dass diese  $\lceil b_R \cdot \log_2(b_R) \rceil$  viele Blockzugriffe zusätzlich kostet.

---

**Aufgabe 7 - 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:

$$\sigma_{B < 7 \cdot 10^3 \wedge B > 7 \cdot 10^3} (R)$$

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  $R[A, B, C]$ ,  $S[A, D, E]$ ,  $T[C, D, E]$  mit folgenden Eigenschaften:

- $|R[A, B, C]| = 1000$  Tupel,  $V(R, A) = 100$ ,  $V(R, B) = 200$ ,  $V(R, C) = 300$
- $|S[A, D, E]| = 4000$  Tupel,  $V(S, A) = 50$ ,  $V(S, D) = 200$ ,  $V(S, E) = 30$
- $|T[C, D, E]| = 2000$  Tupel,  $V(T, C) = 100$ ,  $V(T, D) = 200$ ,  $V(T, E) = 40$

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

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

## Aufgabe 9

1 Punkt

Is the following schedule in a **deadlock state**? Draw the wait-for graph to verify. If it is in a a deadlock state, propose a way to recover from the deadlock.

T1:	T2:	T3:
lock-S(A)		
read(A)		
		lock-X(B)
		lock-S(A)
	lock-X(C)	
		read(A)
lock-X(B)		
	read(C)	
	lock-X(A)	
		lock-S(C)

---

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(A)

          read(B)

          write(C)

                  read(C)

write(B)