

[illegible]

Exercise 1 - *Slotted Page*.1 Point

Consider a slotted page with the following properties:

- Size: $2^{11} = 2048$ bytes
- Addressing mode: **Byte addressing** (individual bytes can be addressed)
- Header: densely packed

The following tuples R , S , T are stored in the slotted page:

- d_1 : $|R| = 9$ Bytes
- d_2 : $|S| = 110$ Bytes
- d_3 : $|T| = 245$ Bytes

How many bytes of the slotted page remain unused (i.e., are still free)?

Exercise 2 - *Extendible Hashing*.

1 Point

The hash function $h(x)$ returns the binary values as shown in the table. **Fill in the missing values and components (global/local depth, pointer).**

Hash table:

x	$h(x)$
6	1100
13	1011
20	1000
21	1010
23	1110
28	1001
33	0010
34	0110

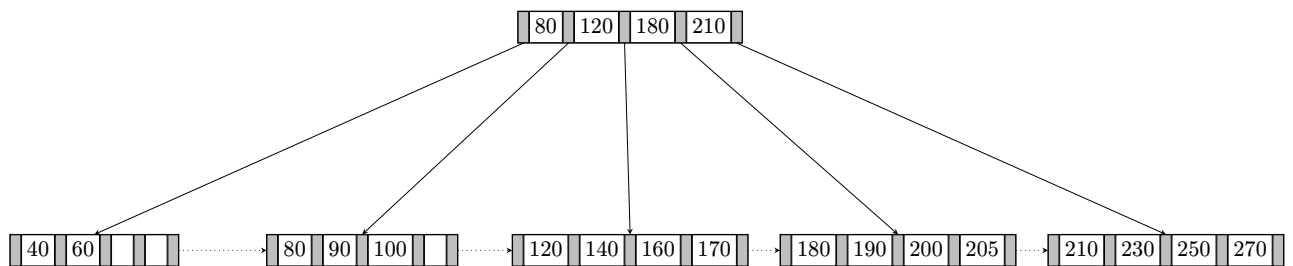
000
001
010
011
100
101
110
111

33
34
20
28
13
21
6
23

Exercise 3 - B^+ -Tree Deletion.

1 Point

Given the following B^+ -tree with $m = 5$. Draw the B^+ -tree after deleting the key **60**.



Exercise 4 - *Duplicates and Indexes.*1 Point

Consider the following (unsorted) relation $R[name]$ with duplicate entries.

- a) Draw a **single-level, non-clustered index** on attribute *name* and resolve duplicates using **buckets**. Hint: Although there are duplicates in the relation, there must not be duplicate search keys in the index.
- b) Duplicates can also be resolved using **tuple identifiers (TID)**. Explain what a **TID** is, and **how to use TIDs** to avoid duplicate search keys in an index.

 $R[name]:$

Noah
George
Noah
Ivy
Harry
Noah
George
Lily
Ivy
Florence
Noah
George

Exercise 5 - *Bitmap Index Scan*.1 Point

Given relation $R[A, B]$ with the following properties:

- $|R| = 10^6$ tuples are stored on $b_R = 10^3$ blocks,
- non-clustering B^+ -tree index on attribute B , $m = 1024$, and maximum height (i.e., each node is minimally occupied),
- duplicates are resolved via tuple identifiers (TID),
- the value 2500 on attribute B is stored 10^3 times in total and distributed among 100 data blocks.

The following query should be answered:

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

Determine the number of **data block accesses** in the **worst case** for

- a. **without using** a bitmap index scan (**0.5 points**),
- b. **by using** a bitmap index scan (**0.5 points**).

For simplicity, you can assume that data blocks that have already been read are not buffered.

Exercise 6 - *Join Algorithms*.1 Point

Given **two relations** R and S with the following properties:

- $R[A]$ with values 1, 2, 3, 4, 5, 6, 7, 8, 9,
- $S[A]$ with values 2, 5, 6, 8, 9, 11, 13, 15, 16, 18, 19, 20,
- a buffer size of $M = 4$ blocks, and
- each block stores 1 tuples.

Draw the partitions of the hash join for both relations along with a step-by-step explanation. The hash function may be chosen freely.

Exercise 7 - *Efficient Query Processing*.1 Point

Consider a relation $R[A, B, C]$ with the following properties:

- $|R| = 10 \cdot 10^{12}$ tuples.
- Each block of data holds $25 \cdot 10^3$ tuples.
- Attribute A holds integer values in the range $[1; 20 \cdot 10^6]$.
- Attribute B holds integer values in the range $[1; 20 \cdot 10^9]$.
- Attribute C holds integer values in the range $[1; 20 \cdot 10^3]$.
- Duplicates are resolved using tuple identifier (TID).
- The following indexes exist:
 - Single level, sequential, sparse index on attribute A with $5 \cdot 10^6$ entries per index block,
 - dense B^+ -tree index on attribute B , $m = 1.024$, minimal height.

The following query is to be processed:

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

State the used **strategy (0.5 points)** and calculate the **number of block accesses (0.5 points)** to process the query **in the most efficient way** (accessing 1 node in the B^+ -tree corresponds to 1 block access).

Exercise 8 - *Join Size Estimation*.1 Point

Consider the following 3 relations $R[A, B, C]$, $S[A, D, E]$, $T[D, E, F]$ and their properties:

- $|R[A, B, C]| = 1000$ tuples, $V(R, A) = 100$, $V(R, B) = 200$, $V(R, C) = 300$
- $|S[A, D, E]| = 4000$ tuples, $V(S, A) = 50$, $V(S, D) = 200$, $V(S, E) = 300$
- $|T[D, E, F]| = 2000$ tuples, $V(T, D) = 200$, $V(T, E) = 400$, $V(T, F) = 600$

Attribute values are assumed to be distributed uniformly and independently. Estimate the size of the following query. $(\sigma_{A=100}(R) \neq \emptyset)$.

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

Exercise 9 - Recoverability.1 Point

Given the following schedule with three transactions T1-T3.

- Is the given **schedule recoverable**? **Justify your answer. (0.5 Points)**
- If **cascading rollbacks** are necessary, identify all transactions for which such a rollback must be performed. If no cascading rollback is necessary, **justify your answer. (0.5 Points)**

T1:	T2:	T3:
<hr/>		
read(X)		
<hr/>		
write(X)		
<hr/>		
	read(X)	
<hr/>		
	write(Y)	
<hr/>		
		read(Z)
<hr/>		
		write(Z)
<hr/>		
		commit
<hr/>		
abort		
<hr/>		

Exercise 101 Point

Consider the following schedule and the **two-phase locking** scheduler.

- (a) Assume that all transactions want to issue a **write** operation on item **A** in the following order: **T3**, **T2**, **T1**. Complete the schedule by inserting lock requests (e.g., **R:lock-S(A)**) and granted locks (e.g., **G:lock-S(A)**) for all remaining **read** and **write** operations. Denote in the schedule if a transaction has to wait since another transaction holds an incompatible lock.
- (b) Draw the wait-for graph after processing all specified lock requests. Does the schedule result in a deadlock? Explain your answer.

T1: **T2:** **T3:**

R:lock-S(A)
G:lock-S(A)
read(A)

read(A)
