Department of Computer and Information Science
Database and Information Systems
Jun.-Prof. Dr. Michael Grossniklaus

Universität
Konstanz

Winter Semester 2013/2014

INF-12950

Database System Architecture and Implementation

Exam, April 16, 2014, 14:00–16:00

Your Student Number

Read the following instructions *before* you start to write your answers.

- The examination consists of 4 questions each of which is worth a total of **25 points**.
- Answers can be given in **German** or **English**.
- You should start the answer to each question on **a new page**.
- Do **not** write in the margins.
- Your answers should be written in **blue** or **black ink** and not pencil. Any parts of your answer that you do not want to be included as part of the final answer should be **clearly scored through** with your pen.
- With the exception of a calculator, **no electronic devices** (laptops, phones, music players, etc.) can be used during the exam.

Do not write below this line

1	2	3	4	Total	Exam	Course	Final

1 Disk, File, and Buffer Management

(a) **Hard Disks.** Consider a disk with a sector size of 512 bytes, 3,876,007 tracks per surface, 216 sectors per track, 7 platters, 14 heads, and an average seek time of 8.5 ms. The track-to-track seek time is 0.25 ms and the disk's mean time between failure is 2,000,000 hours. (7 points)

(i) What is the capacity of a track in bytes?

. . . .

(ii) What is the capacity of each surface?

. . . .

(iii) What is the capacity of the disk?

. . . .

(iv) How many cylinders does the disk have?

. . . .

(v) Which of the following are valid block sizes?

- ☐ 128 bytes
- ☐ 3141 bytes
- ☐ 8192 bytes
- ☐ 25,600 bytes

(vi) If the disk rotates at 15,000 rpm (revolutions per minute), what is the maximum rotational delay?

. . . .

(vii) If one track of data can be transferred per revolution, what is the transfer rate?

. . . .

(b) **RAID Systems.**

(i) What are characteristics of RAID levels 1+0, 3, and 6? (3 points)

. . . .

. . . .

. . . .

. . . .

. . . .

(ii) RAID systems guarantee high data stability by storing so-called *parity bits*. Let us assume a RAID 4 system with four data disks and one parity disk. Which bits are stored on the parity disk for the following bit sequences? (3 points)

Data Disk 1:	00000000
Data Disk 2:	10010010
Data Disk 3:	10101010
Data Disk 4:	00110011
Parity Disk:	. . . .

(iii) Assume we have 10 disks with 750 GB each. How much disk space is available if RAID 5 is used? (2 points)

☐ 3.75 TB

☐ 6.00 TB

☐ 6.75 TB

(c) **Buffer Management.**

(i) The MRU (Most Recently Used) algorithm is a popular replacement policy. Sketch an example of a **worst case** for using MRU. Which alternative policy would you recommend, and why? (5 points)

....

....

....

....

....

(ii) In the following, we assume the existence of 4 buffer pages (1-4) and 5 data blocks (A-E). Initially, all buffer pages are free. Which pages are chosen as victims in **each step** by the MRU algorithm, when loading the following sequence of blocks? (5 points)

A B A B C D E D C B A E A D B C A

....

....

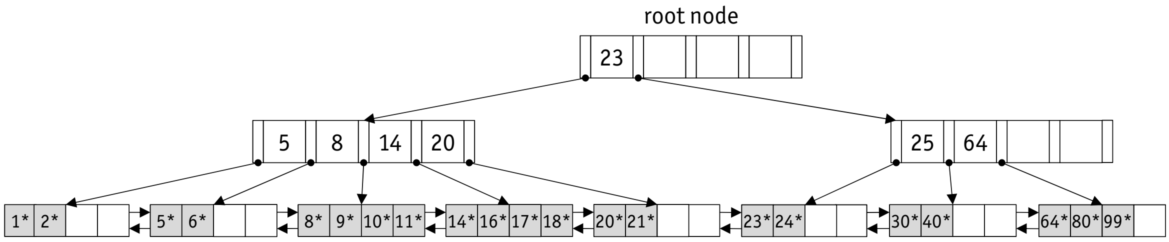
....

....

....

2 Hash and Tree-based Indexes

(a) **B$^+$ Tree.** Given below is a B$^+$ tree with order $m = 2$. For each of the following operations, sketch the resulting tree. Details of the original trees can be omitted as long as all changed nodes are shown. Each operation is to be performed on the **original tree**. (9 points)



(i) Entry **19*** is inserted and siblings are checked for redistribution.

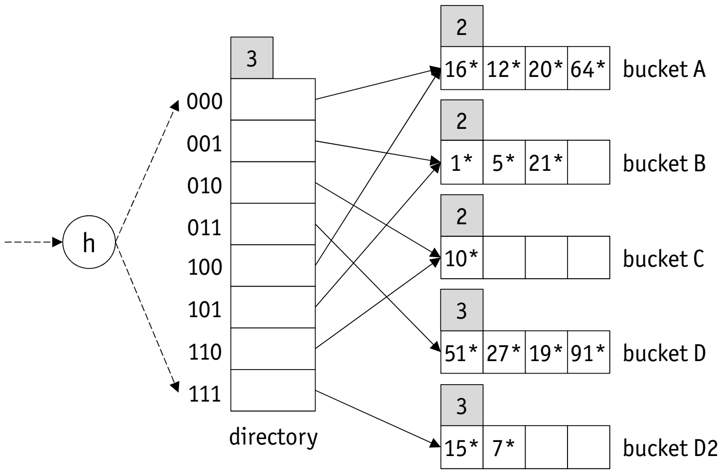
(ii) Entry **19*** is inserted without checking siblings for redistribution.

(iii) Entry **24*** is deleted with checking siblings for redistribution.

Answer

(b) **Extendible Hashing.** Given below is an extendible hashing index with global depth $d = 3$. For each of the following operations, sketch the resulting index. Each operation is to be performed on the **original index.**

(8 points)



(i) Entry **36*** (`0b100100`) is inserted.

(ii) Entries **15*** (`0b1111`) and **7*** (`0b111`) are deleted.

Answer ______

(c) **Linear Hashing.** Given below is a linear hashing index. Assume that the index splits whenever an overflow page is created. For each of the following operations, sketch the resulting index. Each operation is to be performed on the **original index**. (8 points)

level = 0	h_1	h_0					
	000	00	32*	8*	24*		
	001	01	9*	25*	41*	17*	
	010	10	14*	18*	10*	30*	← *next*
	011	11	31*	35*	7*	11*	
	100	00	44*	36*			
	101	01	13*	45*			

(i) Entry **57*** (0b111001) is inserted.

(ii) Entries **13*** (0b1101) and **45*** (0b101101) are deleted.

Answer

3 Query Evaluation

(a) **Sort Operator.** Assuming external merge sort *without* refinements such as blocked I/O, double buffering, or replacement sort is used to sort a file, answer the following questions for each of these scenarios. (15 points)

- A file with 25,000 pages and five available buffer pages.
- A file with 50,000 pages and nine available buffer pages.
- A file with 5,000,000 pages and 65 available buffer pages.

(i) How many runs will be produced in the first pass?

(ii) How many passes will it take to sort the file completely?

(iii) What is the total I/O cost of sorting the file?

(iv) How many buffer pages would be required to sort the file completely in just two passes?

Answer ______________________________

(b) **Join Operator Algorithms.** Suppose a database contains a relation **R** and a relation **S**. There exists a key-foreign key relationship between **S**.*k* (key) and **R**.*fk* (foreign key). Relation **R** has 500 pages with 80 tuples each, whereas relation **S** has 1000 pages with 150 tuples each. The query processor has 250 buffer pages available. Assuming an access time of 8.5 ms per page, compute the total time required to evaluate

$$\mathbf{R} \bowtie_{k=fk} \mathbf{S}$$

for the following three join algorithms. For each join, you may choose which of the two relations is the outer and which is the inner relation.

(10 points)

(i) sort-merge join (with sorted inputs)

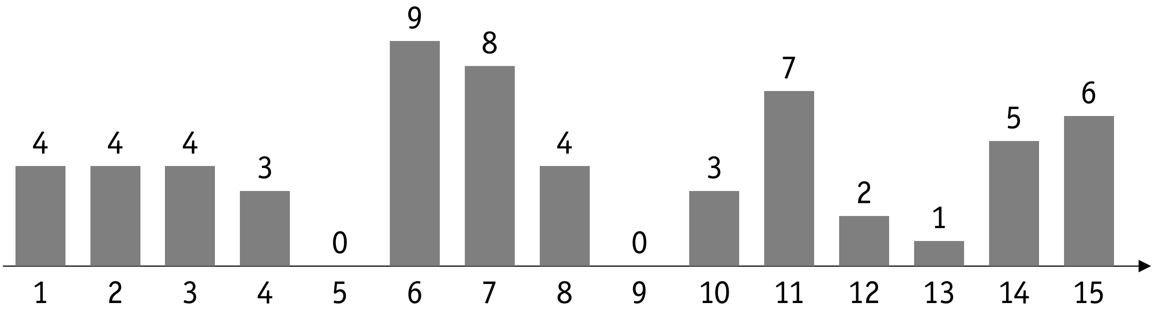
(ii) (grace) hash join (without re-hashing)

(iii) nested loops join (without blocking)

Answer

4 Query Optimization

(a) **Histograms.** Assume a column **R.**a of SQL type INTEGER (domain $\{\dots, -2, -1, 0, 1, 2, \dots\}$) with the following actual non-uniform value distribution in a relation **R.** (10 points)



(i) Create an **equi-width histogram** for **R.**a with width 5.

(ii) Create an **equi-depth histogram** for **R.**a with depth 12.

(iii) Estimate the result cardinality of the following queries based on these two types of histograms. Compare them to the estimated cardinality under the **uniformity assumption** as well as to the **actual value.**

- `SELECT * FROM R WHERE a = 5`
- `SELECT * FROM R WHERE a > 10`
- `SELECT * FROM R WHERE a > 5 AND a < 10`

Answer

(b) **Heuristic and Cost-based Optimization.** Consider the following schema that captures information about pilots, airlines, and aircrafts. (15 points)

> **Pilot**(*pid*: `integer`, *lid*: `integer`, *salary*: `integer`, *rank*: `char(20)`)
>
> **Airline**(*lid*: `integer`, *name*: `char(20)`, *revenue*: `real`, *phone*: `char(10)`)
>
> **Aircraft**(*lid*: `integer`, *code*: `integer`, *payload*: `integer`, *range*: `real`)

Now also consider the following query.

```
SELECT L.name, C.code, C.payload
  FROM Pilot P, Airline L, Aircraft C
 WHERE P.lid = L.lid AND L.lid = C.lid AND
       L.revenue >= 750,000 AND C.range > 3500 AND
       P.salary <= 80,000 AND P.rank = "Co-Captain"
```

(i) Draw a **relational algebra tree** for the above query.

(ii) List all **join orders** in which pairs of relations can be joined to compute the query result. Limit this set of join orders to those that a typical relational query optimizer would consider, i.e., only left-deep plans which do not require the computation of cross-products.

(iii) **Rewrite** (or **transform**) the query to increase the number of join orders considered by the optimizer.

(iv) Based on your relational algebra tree, identify one **query evaluation plan** that reflects the order of operations, the access methods, and physical operators, which a decent query optimizer would choose. Your query evaluation plan does **not** have to be the optimal plan.

(v) In your query evaluation plan, identify an example of **heuristic optimization**.

(vi) In your query evaluation plan, identify an example of **cost-based optimization**.

Answer

Answer