NORTHEASTERN UNIVERSITY

Example Final exam

CS 3200

DR Kathleen Durant

4/19/2013

Problem 1: NOSQL (10 points)

Please compare the ACID relational properties to the BASE NO SQL properties.

Problem 2: True/False, Short open-ended responses, multiple choice (30 points)

2.1 True/False Searching for a value in a B+ tree, search always starts at the root node and moves downwards using the pointers to navigate the tree (2)

2.2 True/False Consider a relation that has one clustered index. When retrieving records for an inequality search, the clustered index should always be used. **Briefly explain your answer**. (6)

2.3 True/False Copying-up during an insertion/deletion of a record in a B+ tree involves moving records between neighboring leaf nodes (blocks) to ensure each leaf block contains at least the minimum number of records of a B+ node(50% capacity). (3)

2.4 True/False. A RAID 5 system of N disks increases both reliability and performance of the N disk system. (3)

2.5 True/False Within the execution of a query plan, materialization is when a temporary, intermediate result set is stored (written out) to secondary storage (disk). (2)

2.6 A page is unpinned in the DBMS' internal buffer pool:

a) By the buffer manager
b) By the transaction who requested the page
c) By the file manager
d) All of the above (4)

2.7 Please describe a search scenario where a clustered hash index is preferable to a clustered B+ tree index. (5)

2.8 Please describe the concepts global depth and local depth within the extendible hashing algorithm. Please give an example where their values would be different. (5)

Problem 3: Extendible hashing and linear hashing insertions (20 points)

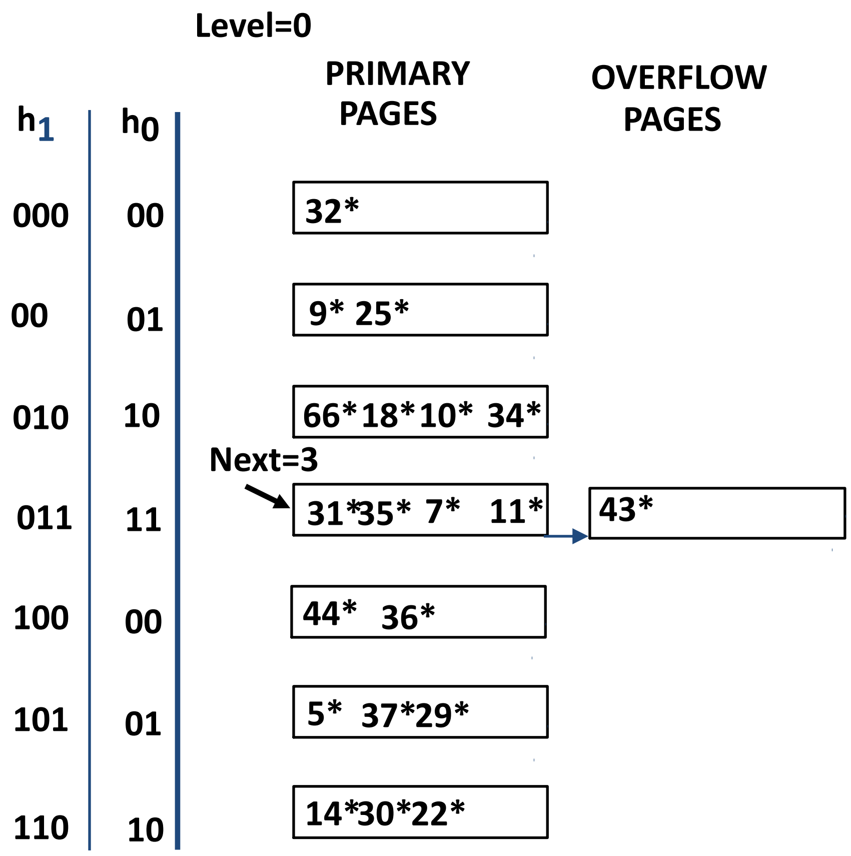
a) Please review the current figures for the extendible hash table and the linear hash table and describe the current state of the hash algorithm represented in each figure. The linear hash state is represented by: current level of the hash function and the next pointer. The extendible hash table state is represented via the local depth of each bucket and the global depth of the directory. (4)

b) Next insert the following records into the extendible hashing and linear hashing tables and draw the resulting hash and data structures after each insertion: h(r1) = 26 (11010) (8) and h(r2) = 27 (11011) . (8)

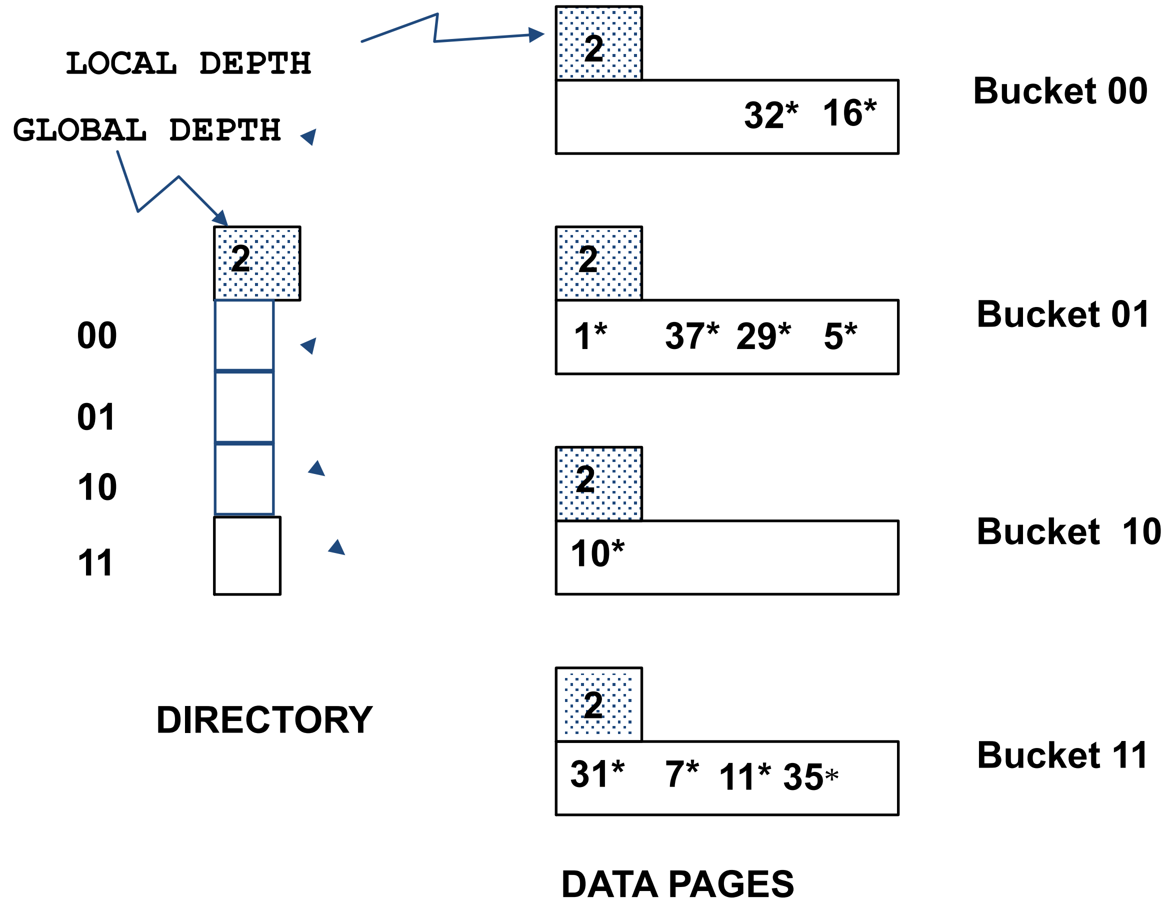
The table below is for reference.

Decimal	Binary	Decimal	Binary	Decimal	Binary	Decimal	Binary
32	100000	10	1010	11	1011	37	100101
9	1001	34	100010	43	101011	29	11101
25	11001	31	11111	44	101100	14	1110
66	1000010	35	100011	36	100100	30	11110
18	10010	7	111	5	010	22	10110

Linear Hash Table



Extendible Hash Table



LOCAL DEPTH

GLOBAL DEPTH

2

00

01

10

11

DIRECTORY

2

32* 16*

Bucket 00

2

1* 37* 29* 5*

Bucket 01

2

10*

Bucket 10

2

31* 7* 11* 35*

Bucket 11

DATA PAGES

Problem 4: External Sort Algorithm. (15 points)

Suppose you have a file with 666 pages and 3 buffers pages to use for sorting the file.

How many runs will you produce in the first pass of the external sort algorithm? (5)

How many passes will it take to sort the file completely? (5)

How many I/O's did it involve? (5)

Problem 5: Cost analysis (10 points)

Relation:

Person(name varchar(28), address varchar(40), title varchar(20), department varchar(20), description(148));

The person table contains 1,024 records. Each data block contains 4096 bytes. Each integer and date field is 4 bytes and each record id is 8 bytes. There is a clustered B+ tree index on pkey. Assume the average time to read a page is 15 milliseconds.

Apply the I/O cost model analysis presented in chapter 8 to approximate the I/O cost for the following query:

select pkey from Person where pkey = 20;

Problem 6: Query Plan (15 points)

Consider the following relational schema representing a database of email messages with a keyword index. Keywords may occur multiple times in an email message; an email message may have multiple keywords within it. Position is the key word's position within the email. Positions are unique for each email.

Email (<u>eid</u> integer, fromuser varchar (100), touser varchar (100), sentdate date, subject varchar (256), body varchar (3636))
KeyWordOccurrence (<u>kid</u> integer, <u>eid</u> integer, position integer)
KeyWord (<u>kid</u> integer, keyword varchar (100));

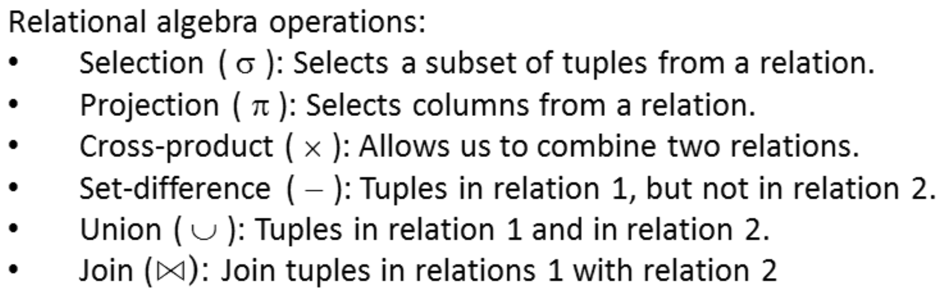
a) Please provide at least 2 different query expression trees for the following query: (5)

Select kid, fromuser from Email JOIN KeyWordOccurrence on Email.eid = KeyWordOccurrence.eid where touser = 'Kathleen' and sentdate > '01/01/2013';

Additional description available to the query optimizer: Each of the three tables has a clustered B+ tree on the primary key of the table. The data block size is 4096 bytes. The Email table has 4096 records, the KeyWordOccurrence table has 20480 records and the KeyWord table has 256 records. The current system has 78 buffer pages it can allocate to this query.

b) Considering the information you have on these tables and the buffer page availability, convert one of your expression trees to a query evaluation plan. (5)
c) Please choose the JOIN algorithm you believe is best suited to address the given constraints. In particular, consider each of the following JOIN algorithms: block nested-loop join, sort-merge join and index nested loop join. Justify your answer. (5)

Resources



Relational algebra operations:

- Selection (σ): Selects a subset of tuples from a relation.
- Projection (π): Selects columns from a relation.
- Cross-product ($\times$): Allows us to combine two relations.
- Set-difference ($-$): Tuples in relation 1, but not in relation 2.
- Union ($\cup$): Tuples in relation 1 and in relation 2.
- Join ($\bowtie$): Join tuples in relations 1 with relation 2