CS 245
Midterm Exam – Winter 2013

This exam is open book and notes. You can use a calculator and your laptop to access course notes and videos (but not to communicate with other people). You have 70 minutes to complete the exam.

Print your name: SOLUTION

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

Signed:_____

Problem	Points	Maximum
1		10
2		10
3		10
4		10
5		10
6		10
Total		60

Problem 1 (10 points)

Consider relations $R(A, B, C)$, $S(C, D)$ and $T(A, B, C)$. The following sub-problems ask you to rewrite relational algebra expressions. You can assume that the relations contain sets (not bags). IMPORTANT: If the requested rewrite is *not* feasible, state so and briefly explain why. Also, make sure there are no useless expressions in your rewrite, e.g., $\pi_{CD}S$ and $\sigma_{A \neq A}R$ are useless.

(a) (3 points) Rewrite the following expression by pushing the projection as far down as possible.

$\pi_D[\sigma_{A=3}(R \bowtie S)]$

Solution:

$$\pi_D(\sigma_{A=3}(\pi_{A,C}(R) \bowtie S))$$

Pushing the Sigma inside the join, but outside the $\pi_{A,C}(R)$, and maybe putting a π_C outside it is fine as well.
1 point deducted if the useless term $\pi_{C,D}(S)$ is present.
2 points deducted if the π is pushed inside partially but not all the way to $\pi_{A,C}(R)$.

(b) (3 points) Rewrite the following expression so there is no union operator and a single selection operator:

$[(\sigma_{A=1}R \bowtie S] \cup [R \bowtie (\sigma_{D=2}S)]$

Solution:

$$\sigma_{A=1 \text{ OR } D=2}(R \bowtie S)$$

(c) (2 points) Rewrite the following expression by pushing the duplicate elimination operator as far down as possible:

$\delta[\sigma_{A=2}(R \bowtie S)]$

Solution:

$$\sigma_{A=2}(\delta(R) \bowtie \delta(S))$$

Pushing the σ inside onto $\delta(R)$ is fine. If you simply mention that the problem involved sets, not bags, and hence the δ operator is unnecessary, that is fine as well.

(d) (2 points) Rewrite the following expression by pushing the join down:

$(R - T) \bowtie S$

Solution:

$$R \bowtie S - T \bowtie S$$

Problem 2 (10 points)

(a) Consider an extensible hash structure where buckets can hold up to two records and no overflow blocks are allowed. Initially the structure is empty. The hash function we use generates $b = 4$ bits total.

Say we insert three records, where the search key of each record generates a distinct 4-bit hash value. (No records are deleted.) We are told that after the three insertions, X buckets have been allocated.

(i) (2 points) What is the minimum possible value of X?

MINIMUM VALUE OF X:_____

Solution: 2

(ii) (3 points) What is the maximum possible value of X?

MAXIMUM VALUE OF X:_____

Solution: 4

(b) (5 points) Now consider a linear hash table (not the extensible hash tables of part (a)). As before, the hash function used generates a $b = 4$ bit key. No overflow bocks are allowed, so when necessary the table is expanded to avoid the use of overflow chains.

The linear hash table is initially empty, and three records are inserted (no deletions). The keys of the inserted records yield distinct hash keys. We are told that after the three insertions, a total of 8 blocks have been allocated.

Give an example of three hash keys that can lead to this behavior. (There is more than one answer, just give one of the possible sequences.)

Hash key of first record (b bits):_____

Hash key of second record (b bits):_____

Hash key of third record (b bits):_____

Solution:
Any 3 distinct 4 bit keys ending with 11 are correct. 0 points if even one of the keys does not end with 11.

Problem 3 (10 points)

Consider a Hard Disk with the following specifications:

- Disk does one full revolution in 512 μsec.
- 4 platters, and 2 surfaces each platter
- Usable capacity: 5 Gigibytes (i.e., 5×2^{30} bytes)
- Number of cylinders: $64 + 128$ (see below)
- 1 block = 4 Kibibytes (i.e., 4×2^{10})
- negligible overhead between blocks (gaps)
- Average seek time: 10,000 μsec

There are 64 inner cylinders and 128 outer cylinders. The outer cylinder have double the density than the inner ones, that is, an outer cylinder has twice the number of blocks than an inner cylinder.

Hint: To compute your answers below, work with powers of 2, and you can leave your answer as a power of 2 if appropriate. For example, if the answer is 3×2^{20} divided by 2^5, just leave your answer as 3×2^{15}.

(a) (2 points) What is the total number of blocks on the disk?

Solution:
5×2^{18}

(b) (1 point) How many of the blocks are on the inner cylinders?

Solution:
2^{18}

(c) (1 point) How many blocks are on the outer cylinders? (Note: your answers for parts (b) and (c) should add up to the total in part (a).)

Solution:
2^{20}

(d) (2 points) On the inner cylinders, how many blocks are on each track?

Solution:
2^9

(e) (2 points) Once the head arrives at the beginning of an inner block, how much times does it take to transfer a block off the disk?

Solution:
1 microsecond

(f) (2 points) What is the expected time to read a block that resides on an inner cylinder? (Include the three types of delays involved.)

Solution:
10257 microseconds

Problem 4 (10 points)

Suppose we are designing a database relation named*Enrollment* to store student enrollments in different courses during a quarter. The table stores basic information about the students and the list of courses in which the student has enrolled. For simplicity, let us assume that a student cannot be enrolled in more than 5 courses in a quarter. Here is the description of database column fields:

- RecordHeader CHAR(10)
- StudentID CHAR(9)
- Name CHAR(30)
- Gender CHAR(1)
- Variable number of course names (max 5) each of type CHAR(30)

Let the size of a block be 4096 bytes out of which 96 bytes are used for the block header. Records are not spanned.

(a) Assuming that we used fixed-length records to store the tuples, what is the length (in bytes) of each record and what is the number of such tuples that can be stored per block.
(2 points) SIZE OF A RECORD:_____

Solution:
200 bytes
(2 points) MAXIMUM NUMBER OF RECORDS:_____

Solution:
20

(b) Using fixed-length records to store tuples is definitely going to result in space wastage. To be more space efficient, we are now going to use variable-length records to store the repeating course fields. We will still use 30 bytes to store each course name. The other fields are still of fixed length. However, we now add one byte to the record header to store the record length (in bytes).

Assuming that a student has 3 courses on the average, how many records can we store in a block, on average? Of course, some blocks will hold fewer than this average number of records. In the worst case, what is the smallest number of records that a *full* block may be holding?

(3 points) AVERAGE NUMBER OF RECORDS:_____

Solution:
Size of an average record = 141
Average number of records = 4000/141, rounded to 28.

(3 points) WORST CASE MIN NUMBER OF RECORDS IN FULL BLOCK:_____
Solution:
Maximum size of a record = 201
Minimum number of records = 4000/201, rounded to 19.

Problem 5 (10 points)

Consider a B+ tree of order n. To simplify this problem, assume n is an odd integer.

(a) (2 points)What is the minimum number of records the tree can index when it has k levels? Assume that $k > 1$. (Provide the answer in terms of k and odd number n.)

MINIMUM NUMBER OF RECORDS : _____
Solution:

$$2 * ((n+1)/2)^{k-1}$$

(b) (2 points)What is the maximum number of records the tree can index when it has k levels? Assume that $k > 1$. (Provide the answer in terms of k and odd number n).

MAXIMUM NUMBER OF RECORDS : _____
Solution:

$$n * (n+1)^{k-1}$$

(c) (3 points) You are told that your system uses a B+ tree with nodes of size $n = 5$, and that it currently has 2 levels. (We are using small numbers to simplify any arithmetic you may need to do.) Say there are currently 20 keys indexed. You may assume that there are no duplicate keys.

At this point, say we start inserting additional keys into the tree, and after Y insertions, the number of levels changes to 3. What is the *largest possible* value of Y? That is, there must be some scenario for which it takes this maximum Y insertions (starting at 20 keys) to increase the number of levels to 3.

LARGEST POSSIBLE Y: _____
Solution:
Using part (b), maximum number of records in the tree with level 2 is 5 * (5+1) = 30. Hence, there can be a maximum of 11 more insertions.

(d) (3 points) Under the same scenario of part (c), what is the *smallest possible* value of Y? That is, there must be some scenario for which it takes this minimum Y insertions (starting at X records) to increase the number of levels to 3.

SMALLEST POSSIBLE Y: _____
Solution:
Using part (a), minimum number of records in the tree with level 3 is

$$2 * ((5+1)/2)^2 = 18. \tag{0.1}$$

Hence, the number of levels can change in the next insertion.

Problem 6 (10 points. 1 point for each)

State if the following statements are true or false. Please write TRUE or FALSE in the space provided.

(a) Solid state disks (SSDs) provide block access to data, just like a magnetic hard drive.

ANSWER:_____
Solution:
True

(b) The "containment of value sets" property states that the result of a selection query on relation R produces a subset of R tuples.

ANSWER:_____
Solution:
False

(c) Dense indexes are better than sparse indexes when most queries are searching for keys that do not exist in the relation.

ANSWER:_____
Solution:
True

(d) The answer to a nearest-neighbor query with respect to point p is the point closest to p returned by any range query centered on p with a non-empty answer set.

ANSWER:_____
Solution:
True

(e) Encrypting records in a database makes it harder to build indexes for them.

ANSWER:_____
Solution:
True

(f) Social security numbers definitely do not follow a Zipfian distribution.

ANSWER:_____
Solution:
True

(g) A column store is superior to the more traditional row store when queries refer to many attributes (columns).

ANSWER:_____
Solution:
False

(h) A grid index can only be used when the values to index come from a countable domain. (For example, real numbers are not countable.)

ANSWER:_____
Solution:
False

(i) The maximum seek time occurs when the heads have to move across half of the tracks.

ANSWER:_____
Solution:
False

(j) A hash-join algorithm must use two different hash function, one of each of the relations being joined.

ANSWER:_____
Solution:
False