CS 245
Midterm – Summer 2009

This exam is open book and notes. You have 90 minutes (1 hour, 30 minutes) to complete it.

Remote students: Your proctor should scan and email the completed exam to cs245-sum0809-staff@lists.stanford.edu.

Print your name:_____

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)

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

(a) Doubling the rotation speed of a disk also doubles the transfer rate.

ANSWER:_____

(b) Replicating data on separate disks can improve the seek time when searching the disks concurrently for the same file.

ANSWER:_____

(c) The larger gaps are on a disk track, the smaller the burst bandwidth will be.

ANSWER:_____

(d) The LRU (least recently used) buffer replacement strategy is good when repeatedly scanning a relation.

ANSWER:_____

(e) It is easier to compress data when using column stores.

ANSWER:_____

(f) Variable format records can be fixed length.

ANSWER:_____

(g) A grid file index can support range queries, partial-match queries, and nearest-neighbor queries well, as long as data is fairly uniform in distribution.

ANSWER:_____

(h) In a B+tree, all blocks are between half-full and completely full at all times.

ANSWER:_____

(i) Extensible hashing does not have overflow buckets.

ANSWER:_____

(j) Estimates for final result sizes are always the same for equivalent logical query plans.

ANSWER:_____

Problem 2 (10 points)

Consider an indexed sequential file consisting of 10,000 blocks. Each block contains 10 fixed sized records. Each key value found in the file is unique. For this problem, assume that:

- Pointers to blocks are 10 bytes long.
- Pointers to records are 20 bytes long.
- Index blocks are 5000 bytes (in addition to the header).
- Search keys for file records are 10 bytes long.

(a) Assume that the file and index blocks are stored contiguously on the disk. How large (in blocks) would a sparse one-level, primary index be? Design the index so it would use the minimum amount of space.

Number of blocks:_____

(b) Assume that the file blocks are *not* contiguous on disk. How large (in blocks) would a sparse one-level, primary index be? Design the index so it would use the minimum amount of space.

Number of blocks:_____

(c) Suppose you now construct a one-level, dense secondary index. Compute its minimum size in blocks.

Number of blocks:_____

Problem 3 (10 points)

Consider a linear hashing storage system. Each block can hold 4 records. Initially, m is set to 0, and $i = 1$. (We use the m and i notations in the slides.) The following records have been stored:

(A) Key hashes to 0111

(B) Key hashes to 1111

(C) Key hashes to 0001

(D) Key hashes to 0111

(E) Key hashes to 1000

(F) Key hashes to 1110

(G) Key hashes to 0011

(H) Key hashes to 0010

(I) Key hashes to 0000

(J) Key hashes to 0011

(K) Key hashes to 0100

(L) Key hashes to 0001

(a) Simulate inserts of the keys in the order they are listed where m increases when the utilization $U > 1$. We use the utilization defined in class, which is the number of used slots over the total number of slots in the hash table (excluding those in overflow buckets). For your answer, only show the final state of the linear hash structure for these keys using the diagram like the one we used in class.

(b) Suppose that m increases whenever there is an overflow bucket. Only show the final state of the linear hash structure for the same sequence of keys.

(c) What is the utilization for the final linear hash structure in (b)?

Utilization:_____

Problem 4 (10 points)

Consider a query optimizer that uses histograms. In particular, the following information is known about the attribute A of relation R where R has the attributes $R(A, B, C)$. Attribute A is of type integer.

- There are 100 tuples with A values between 1 and 10.
- There are 300 tuples with A values between 11 and 20.
- There are 200 tuples with A values between 21 and 30.

(a) Suppose that for each range, the possible A values of tuples are uniformly distributed. For example, a tuple with an A value within the range of 21 and 30 has one of the 10 possible values with equal probability. How many tuples are expected in the result of the query $\sigma_{A=7}(R)$?

Expected number of tuples:_____

(b) Now suppose for the same R we have the following information.

- There are 3 unique A values (uniformly distributed) within the A values between 1 and 10.
- There are 5 unique A values (uniformly distributed) within the A values between 11 and 20.
- There are 10 unique A values (uniformly distributed) within the A values between 21 and 30.

How many tuples are expected in the result of the query $\sigma_{A=17}(R)$ when 17 is one of the 5 unique values for the second range?

Expected number of tuples:_____

(c) Using the same R from (b), now consider the query $R \bowtie_{A=D} S$, where S has attributes $S(D, E, F)$. Assume that the following information is known about the attribute D of relation S. Attribute D is of type integer.

- There are 50 tuples with D values between 1 and 10. In this range, there are 5 unique D values uniformly distributed.
- There are 100 tuples with D values between 11 and 20. In this range, there are 5 unique D values uniformly distributed.
- There are 200 tuples with D values between 21 and 30. In this range, there are 5 unique D values uniformly distributed.

Assuming that the containment of value sets assumption holds when joining R and S, how many tuples are expected in the answer?

Expected number of tuples:_____

Problem 5 (10 points)

We want to join two relations R and S. Relation R has 20,000 tuples, S has 45,000 tuples, 25 tuples of R fit into one block, and 30 tuples of S fit into one block. Each relation is stored contiguously on the disk. Relation R is sorted on the join attribute. Related S is unsorted but has a B+ tree index on the join attribute. Assume that the index is permanently in memory. In addition, there are 30 memory buffers available for the join.

(a) Say the join is performed with the most basic sort-merge join presented in class. (The B+ tree is not used.) How many IOs are required to execute this query? (Remember not to count in any part of this problem the IOs needed to write the final answer.)

Required number of IOs:_____

(b) Consider a variation of the merge join algorithm that scans the join attribute of S through the index, resulting in the S join values being retrieved in sort order. Assume that each R tuple joins on average k tuples from S, $k < 1$. How many IOs is the join expected to take (as a function of k).

Expected number of IOs:_____

(c) For what values of k will strategy (b) (use index) be more efficient than strategy (a)?

Values of k:_____

(d) Assume that k can be larger than 1. In the worst case, how many IOs will strategy (b) (use index) take?

Worst case IOs:_____

Problem 6 (10 points)

Consider a B+ tree with order n.

(a) Suppose this B+ tree has a height of j and is full (i.e., all nodes are at maximum capacity). How many new records must be inserted for the tree to be full with a height of j+1?

Number of records:_____

(b) If the B+ tree points to r records, what is the *minimum* height j possible? Express your answer as a function of n and r.

Minimum height:_____

(c) If the B+ tree points to r records, what is the *maximum* height j possible? Express your answer as a function of n and r.

Maximum height:_____