CS 245
Midterm Exam – Winter 2009

This exam is open book and notes. You have 70 minutes to complete it.

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)

Suppose we have a 8192-byte block where we store 100-byte records. Suppose the block header only contains an offset table using 2 byte pointers to records within the block. Starting with an empty block, we repeatedly

- Insert 2 records,
- Delete 1 record,

until there is no more space in the block. When a record is deleted, its pointer in the block header is replaced by a tombstone. What is the maximum number of records this block will hold? How much free space will there be in the block at this point?

MAX NUMBER OF RECORDS:________________

FREE SPACE:________________

Problem 2 (10 points)

You need to create a conventional index on attribute B of relation $R(A, B)$. Attribute A is the primary key and B is a non-unique attribute. Relation R is stored sequentially based on the A attribute. Attribute A is 10 bytes and B is 40 bytes, so each R tuple is 50 bytes long. All pointers (regardless of their type) take 10 bytes.

You are considering the following two options for the conventional index:

- Full: The index contains all (key, record pointer) pairs.
- Indirection: The index contains (unique key, pointer) pairs, where each pointer links to an indirection file (bucket) containing the record pointers.

Based on parameters $T(R)$, $V(R, A)$ and $V(R, B)$, what is the size of the index in each case? Briefly explain your answer. How should T(R) be related to V(R,B) to make the indirection method more space-efficient than the full index implementation without indirection?

Num. Bytes for Full Option:________________

Num. Bytes for Indirection Option:________________

When is Indirection more Space-efficient?:________________

Problem 3 (10 points)

Consider an extensible hash structure where buckets can hold up to two records and no overflow blocks are allowed. Initially the structure is empty.

(a) Simulate inserts of the following keys in the order they are listed. Show the extensible hash structure for these keys using the diagram like we used in class and homeworks.

```
0001
0000
1000
0010
1111
0011
1100
1110
```

(b) Now suppose we execute the following deletes on the same table: 1100, 1110. Show the extensible hash structure at the end of these steps. Assume that deletions re-structure the extensible hash table (i.e. we merge buckets when possible).

(c) For this part, ignore the previous inserts and deletes. (Buckets can still hold up to two records and no overflow blocks are allowed.) We start with an empty extensible hash table with a directory that has 2 entries. After some insertions (and no deletions), we are told that the directory has grown to 512 entries.

(i) What is the minimum number of keys that this hash table can hold? Give a sample key sequence that would generate this worst case behavior for such extensible hash table (you can assume keys are 9 bits long).

(ii) If the table holds the minimum number of keys, what is the minimum number of buckets that have been allocated (assuming there are no deletes)?

Problem 4 (10 points)

You are building a B^{+} tree for a system with the following parameters:

- key field has V bytes
- block size is B bytes
- record pointers are P_r bytes
- block pointers are P_b bytes

Also, assume that the index is dense, i.e., leafs point to records that are stored elsewhere.

(a) Let n_l be the maximum number of keys that a leaf node could have in this setting. Express n_l as a function of parameters V, B, P_r and P_b.

$n_l =$:________________

(b) Let n_n be the maximum number of keys that a non-leaf node could have in this setting. Express n_n as a function of parameters V, B, P_r and P_b.

$n_n =$:________________

(c) Should you actually allow n_l and n_n keys in leaf and non-leaf nodes respectively? Or should you pick one (or the other value) and use it for all nodes? Discuss the tradeoffs.

Using same n is desirable because:________________

Using different values is desirable because:________________

(d) Assume that $n_l = n_n = 100$. What is the maximum number of records that the B$^+$-tree can point to , if it has 3 levels (i.e. root, one intermediate level and the leafs)? How many blocks do you need for the storage of the index in this case?

Maximum number of records:________________

Required blocks:________________

Problem 5 (10 points)

Consider a linear hash table that is initially empty. Each block has space for at most 2 records.

You insert records with the following hash keys, in this order:

1. 00000
2. 00001
3. 00010
4. 00011
5. 00100
6. 00101
7. 11000
8. 10000

As you insert records, you increase parameters i and m as necessary to avoid any overflow chains. (Note that this strategy is different from what we discussed in class. In class we discussed changing parameters when the utilization increased above a threshold. Instead, in this problem we change parameters to avoid any overflow chains.) Furthermore, you only increase these parameters when necessary to avoid overflow chains.

(a) Show the state of the linear hash table after the *first six* records have been inserted. Specify the value of i and m are this point. (Use notation from the lecture notes.)

(b) Show the state of the linear hash table after the *all* eight records have been inserted. Specify the values of i and m are this point. (Use notation from the lecture notes.)

Problem 6 (10 points)

Consider a relational DBMS that has two relations: $R(A, B, C)$ and $S(D, E, F)$. Suppose we have the following query and are trying to estimate the number of tuples in the result.

```
SELECT * FROM R,S
WHERE R.A = S.D AND R.B = S.E AND R.A = a
```

We have the following statistics and assumptions:

- T(R) = 1000
- V(R,A) = 10
- V(R,B) = 20
- T(S) = 2000
- V(S,D) = 50
- V(S,E) = 100
- Containment of value sets
- Preservation of value sets

(a) What is the estimated number of tuples in the result assuming we push the selection `A=a` down as much as possible?

Number of Tuples:________________

(b) What is the estimated number of tuples when we do the selection `A=a` as late as possible?

Number of Tuples:________________

(c) Are the numbers in (a) and (b) the same? If so, are they always the same for any statistics?