CS245 Midterm Exam – Summer 1999

This exam is open book and notes. You have 90 minutes to complete the exam. There are four parts in the exam with a total of 90 points. Please read the instructions for each part carefully before starting to work on the questions. In Parts 1–3, if you wish to explain or demonstrate your solution to a problem for partial credit, you may use the space next to the question or the back of the pages (and warn us on the front). Please use this option sparingly, e.g., if you think the question is flawed or open to multiple interpretations, because we shall only be awarding partial credit in rare situations.

Print your name: ______________________________

In accordance with both the letter and spirit of the Honor Code, I have neither given or received assistance on this examination.

Signed: ______________________________

Part	Maximum	Points
I	16	
II	12	
III	47	
IV	15	
Total	90	

Part I. (16 points) Indicate whether each of the following statements is TRUE or FALSE. If you chose FALSE, please correct it or briefly explain why.

1. The LRU algorithm should not be used for the buffer management of B+-tree indexes.

Ans: ______________

2. Since a plain B-tree can store record pointers in the inner nodes, it is more compact than a B+-tree and incurs lower search cost.

Ans: ______________

3. Consider relations $R(A, B)$ and $S(B, C)$ where $n_R = 2000$, $n_S = 1000$, and B is a key for S. The expected number of tuples in $R \bowtie S$ is less than or equal to 1000.

Ans: ______________

4. Split records is a useful technique for organizing data records with a hybrid format.

Ans: ______________

5. The time for a request to wait to be serviced may be longer using the elevator algorithm than using the first-come-first-serve algorithm.

Ans: ______________

6. The merge join algorithm should be used for a natural join operation only when the joined relations are sorted, because in cases where the joined relations are not sorted the hash join algorithm is faster.

Ans: ______________

7. It is always better to store a sequential file in as few blocks as possible, because that can reduce the disk I/O's when the file is accessed.

Ans: ______________

8. Any relational algebra expression consisting of select (σ), project (π), and join ($\bowtie$) operators can be transformed to the form $\pi_A(\sigma_C(R_1 \times \cdots \times R_n))$, where A is an attribute list, C is a selection condition, $\times$ is the cross-product operator, and R_i's are the operand relations.

Ans: ______________

Part II. (12 points) Select one correct answer from the listed multiple choices. Points = 4 $\times$ number right $-$ 2 $\times$ number wrong.

1. Let $R(A, B)$, $S(B, C)$, and $T(A, B, C)$ be three relations. Consider the following relational algebra expressions.

E1: $\pi_{A,C}(\sigma_{B<10}(R) \bowtie S - T)$
E2: $\pi_{A,C}(R \bowtie \sigma_{B<10}(S) - \sigma_{B<10}(T))$
E3: $\pi_{A,C}(\pi_A(R) \bowtie \sigma_{B<10}(S) - T)$

Which expressions are equivalent to each other?

(a) E1, E2, and E3
(b) E1 and E2
(c) E2 and E3
(d) E1 and E3
(e) none **Ans:** ____________

Questions 2–3 below refer to two relations $R(A, B, C, D)$ and $S(B, E)$. R contains 5,000 tuples stored in a sequential file sorted on attribute A. S contains 100,000 tuples stored in a sequential file sorted on attribute B. Each block can hold 10 R or S tuples.

2. Which one of the following indexes is most appropriate for answering the query $R \bowtie S$, assuming we can always store the index in memory.

(a) A dense index on R.B
(b) A sparse index on R.B
(c) A dense index on S.B
(d) A sparse index on S.B
(e) A sparse index on R.A **Ans:** ____________

3. Which query benefits the least from a multi-key index on R with A as the first attribute and B as the second attribute?

(a) `select * from R where A = 10`
(b) `select * from R where B = 20`
(c) `select * from R where A < B`
(d) `select * from R where A < C`
(e) `select * from R where C < 100 and A = 10` **Ans:** ____________

Part III. (47 points) Write your final answer for each question.

Questions 1–2 below refer to a disk with an actual (formatted) capacity of 1 gigabyte. The disk has 10 surfaces and 1000 tracks per surface. The disk rotates at 6000 rpm. The average seek time is 15ms. The block size is 8KB.

1. Suppose that we are reading a file F that occupies exactly one entire track, and we want to estimate how long it takes to read the whole file sequentially. What key parameter is missing for doing this? Select one from the following choices:

(a) the diameter of the disk surface — d
(b) the average rotational latency — r
(c) the amount of overhead per track — v
(d) none **Ans:** ____________

Based on your choice above, how long does it take to read the file? You may use the symbol (after each choice) to represent the parameter that you think is useful in the estimation result.

Ans: ____

2. Now, suppose that we want to estimate how long it takes to read a block. What key parameter is missing for doing this? Select one from the following choices:

(a) the diameter of the disk surface — d
(b) the average rotational latency — r
(c) the amount of overhead per track — v
(d) none **Ans:** ____________

Based on your choice above, how long does it take to read a block? You may use the symbol (after each choice) to represent the parameter that you think is useful in the estimation result.

Ans: ____

3. Consider a B+ tree index of order n over a file with r records. What is the minimum number of nodes that we may have to examine when searching for a record using its key value? (Note: Here, we use the notion of "order n" as introduced in the class.)

Ans: ____

4. Suppose that you are designing a file system for a movie database. Each movie record has 10 fields that always occur (e.g., title, year) and 30 optional fields that may or may not be relevant or known for a movie (e.g., special effect designer). Each field has a fixed size of 20 bytes. Assume that each optional field is relevant for a particular movie with probability p.

You are considering two options for the file records:

1. A fixed-format record
2. A variable-format record where all fields are tagged using a 2-byte tag.

For what range of p values is the fixed format option better than variable format option according to the expected storage requirement? (Hint: You may need to estimate the expected size of a record for each option.)

Ans: ________________________________

5. Consider a relation $R(A, B)$ stored in a partitioned hash table with 1024 buckets. Suppose that 20% of the queries on R have the form `SELECT * from R where A =` a, and 80% of the queries have the form `SELECT * from R where B =` b where a and b are various constants. Further suppose that a hash function h_1 produces a string of n_A bits based on the A value, and another hash function h_2 produces a string of n_B bits based on the B value. The combination of the two strings form the hash key used to index the hash table. What should be the values of n_A and n_B in order to minimize the average number of buckets that must be accessed to answer a query?

Ans: ________________________________

6. Let $R(A, B, C)$ and $S(C, D, E, F)$ be two relations. Transform the following SQL query to an equivalent relational algebra expression using only select (σ), project (π), and join ($\bowtie$) operators, in which selections and projections are performed as early as possible.

```
select A, E
from R, S
where R.C = S.C and
      R.A < 10 and
      R.C + S.D > 100
```

Ans: ________________________________

7. Consider a relation $R(A, B, C)$ clustered based on the A value. Please match each query with a structure that is most appropriate for the query. For example, if you think that i_1 is most appropriate for q_1, write q_1–i_1 in the answer. There may be queries that do not need to use any of the given indexes, and there may be indexes that are not useful for any of the given queries. You may leave such queries or indexes unmatched.

queries	**indexes**
q_1. `select * from R where R.A = 10`	i_1. grid index on A and B
q_2. `select * from R where R.B > 20`	i_2. B+ tree dense index on B
q_3. `select * from R where R.A < 10`	i_3. sparse index sequential file on A
q_4. `select * from R`	i_4. extensible hashing on B
q_5. `select * from R where R.A < 10 and R.B > 20`	i_5. partitioned hashing on A and B
q_6. `select * from R where R.A = 10 and R.B = 20`	i_6. linear hashing on A

Ans: ____________________

Questions 8–9 below refer to the following information. You want to build a dynamic hash table for a file with 4096 records. Suppose that all record keys are unique, and you are using a hash function that hashes all these keys to distinct 15-bit hash values. Each disk block can hold 8 records.

8. If you are using an extensible hash table, what are the minimum and maximum possible sizes of the hash directory?

Ans: ____________________

9. If you are using a linear hash table with a reorganization threshold of 6, what are the minimum and maximum numbers of buckets (non-overflow blocks) in the hash table?

Ans: ____________________

Part IV. (15 points) Answer the following questions and **briefly** show your reasoning steps.

Consider two relations $R(A, B, C)$ and $S(C, D, E)$ with the following statistics:

- $n_R = 20,000$, each block can hold 20 R tuples.
- $n_S = 2,000$, each block can hold 10 S tuples.
- $V(C, R) = 10,000$.
- $V(C, S) = 500$.

Suppose that R is sorted on A attribute, S is sorted on C attribute. You have a B+-tree index of order 100 for each relation on the C attribute, which contains pointers to all tuples of the relation. Further suppose that none of the indexes or the relations is stored in memory initially. You are asked to design a physical query plan for the following query:

```
select A from R, S where R.C = S.C
```

(1) Assuming that you have a 4-block memory buffer, what is the best physical query plan you can think of for the above query? Please briefly describe how you are going to use the indexes and the memory buffer.

(2) Estimate the join result size and the worst case query cost (# of block I/Os) for your plan. (You may make the simplifying assumption of *containment of value sets* as described in the class, and you may assume that the C values are uniformly distributed in R and in S.)