**NATIONAL UNIVERSITY OF SINGAPORE**
SCHOOL OF COMPUTING
SEMESTER 2 (2018/2019) MID-TERM EXAMINATION FOR

**CS3223: DATABASE SYSTEMS IMPLEMENTATION**

March 2019 Time Allowed: 60 minutes

NAME: MATRIC NUMBER:

This paper contains 2 Sections. Section A has 10 multiple choice questions. Pick the **BEST** answer (only ONE answer) for each question. **Write your answers in the table given.** You should read the questions in the order given (as some questions are related). Section B has 4 open questions.

**Section A (10 Marks)**

| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |

1. Consider a non-empty B+-tree that adopts FORMAT 2 (i.e., leaf level stores key-pointer pairs). If the order of the tree is d, which of the following statements is/are TRUE?
   (a) The minimum height possible is 1
   (b) The minimum number of data entries is 2d
   (c) For a 3-level B+-tree, the minimum number of data entries is 2d(d+1)
   (d) For a 4-level B+-tree, the minimum number of data entries is $2d(d+1)^2+1$
   (e) (a) and (c)
   (f) (b) and (c)
   (g) (a) and (d)
   (h) (b) and (d)

2. Consider a data file with 1,000,000 (1 million) records. Each record is 100 bytes. The primary key is 4 bytes. We would like to build a B+-tree for the data file on the primary key. Let each index/data page be 4k (4096) bytes, and pointers be 4 bytes. Suppose we adopt FORMAT 1 for the B+-tree (i.e., data are stored at the leaf nodes). Assume that leaf nodes are 100% full. What is the order of the tree?
   (a) 20 (b) 40 (c) 255 (d) 256 (e) 511

3. Referring to Question 2, the height of the tree is
   (a) 1 (b) 2 (c) 3 (d) 4 (e) 5

4. Consider a hard disk with the following specifications.
   3840 RPM
   10 platters, and 2 surfaces each platter
   Usable capacity: 10GB
   Number of cylinders: 256
   1 block = 4 KB
   20% overhead between blocks (gaps)
   Track-to-track seek time: 2 ms
   Average seek time: 20 ms

   There are X blocks per track, where X is
   (a) 128 (b) 256 (c) 512 (d) 1024 (e) 4096

5. Referring to Question 4. Which of the following statements is/are TRUE?
   (a) The average rotational delay is approximately 7.81ms
   (b) The transfer time of reading a block without a gap is approximately 0.02ms
   (c) The transfer time of reading a block with a gap is approximately 0.03ms
   (d) The average time to access a block (without the gap) is approximately 27.83ms
   (e) All of the above
   (f) None of the above

6. Referring to Question 4. Suppose we store a 2-level B+-tree (Format 2) on 2 consecutive cylinders of the disk. Furthermore, suppose the data file is also stored on the consecutive cylinder after the leaf level. For example, root level of the tree is at track X, level 2 (leaf level) is at track X+1, and the data is stored at track X+2. What is the average access cost to perform a single record retrieval?
   (a) 27.49ms
   (b) 47.49ms
   (c) 83.49ms
   (d) None of the above

7. Referring to Question 6. What is the average access cost to perform a range query that requires accessing 2 adjacent leaf nodes, and 10 data pages that are randomly scattered across the track.
   (a) 117.99
   (b) 135.99
   (c) 288.99
   (d) 455.99
   (e) None of the above

8. Insert the following keys (in binary) into an initially empty Linear Hash table in the order given: 1111, 1100, 0011, 0000, 1010, 1011, 1110, 0001. Suppose each bucket contains 2 records, and a split occurs whenever a bucket overflows. Suppose the first bucket is labelled as 0 (as in the lecture), where is the position of the next pointer at the end of the insertion?
   (a) 0 (b) 1 (c) 2 (d) 3 (e) 4 (f) 5 (g) 6

9. Which of the following statements is TRUE? For hash-based techniques, we do not consider ordering within primary and overflow buckets.
   (a) Linear Hashing is order independent
   (b) B+-tree is order independent
   (c) Extensible Hashing is order independent
   (d) Static Hashing is order independent
   (e) (a) and (b) and (d)
   (f) (a) and (c)
   (g) (c) and (d)
   (h) None of the above
   (i) All of the above

10. Which of the following join algorithms can always operate with just 3 buffer pages (You should account for both input and output pages)?
   (a) Block nested-loops join
   (b) Sort-merge join
   (c) Hash join
   (d) (a) and (b)
   (e) (a) and (c)
   (f) (b) and (c)
   (g) All of the above.
   (h) None of the above.

Section B (10 marks)

1. (4 marks) Consider sorting a file R. R has 30 pages, and each page can hold only one record. Suppose we have 5 buffer pages.
   (a) (1 mark) What is the I/O cost to sort this file using the external sort algorithm (that uses internal sort)?

   (b) (1 mark) Suppose we adopt replacement selection instead of internal sort. Suppose using replacement selection results in sorted runs of twice the size of a run (compared to the internal sort above). Note that the last run is the exception and may be smaller. What is the I/O cost to perform the sort? You can ignore the output buffer for generating the sorted runs.

   (c) (2 marks) What is the number of sequential and random I/Os for generating the sorted runs in part (b)? For simplicity, we assume that pages of a run or a file is always stored in the same track/cylinder sequentially. Moreover, assume that different runs or files are always stored on different tracks/cylinders.

2. (2 marks) Insert the following keys into an Extensible Hash table: 1111, 0000, 0111, 0011, 1011. Each bucket can hold only 2 records. Show the final structure.

3. (1 mark) Consider the sort-merge join between tables R and S where |R|<|S|. Suppose both tables are already sorted. What is the minimum cost (best case) in terms of number of I/Os that we can possibly get?

4. (3 marks) Consider the query: SELECT R.* from R, S WHERE R.a = S.b

   Let |R| = 500, ||R|| = 10000, |S| = 1000 and ||S|| = 10000. Suppose there is an index on R.a and another index on S.b. For simplicity, suppose the indexes are unclustered, and both indexes have height of 3 (i.e., leaf nodes are at level 3). Moreover, assume that all R.a values are unique in R, and all S.b values are unique in S. We also assume that only 50% of the tuples match, i.e., 50% of the R.a values appear in S.b. What is the (optimal) cost to process this query under the following three schemes:
   (a) Using nested index join where R is the outer table.
   (b) Using nested index join where S is the outer table.
   (c) Using both indexes to compute the join. For simplicity, suppose each index has 100 leaf nodes.

~~~~~~~~~~~~~~~~~~~~~~~~~~ End of Paper ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~