NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING
SEMESTER 2 (2020/2021) MID-TERM EXAMINATION FOR

CS3223: DATABASE SYSTEMS IMPLEMENTATION

March 2021 Time Allowed: 75 minutes

NAME: MATRIC NUMBER:

This paper contains 20 questions. 5 of these questions are 2-mark questions (the marks are indicated with the questions), while the remaining 15 questions are 1 mark each. The total marks for this paper is 25 marks. For multiple-choice questions, pick the **BEST** answer (only ONE answer) for each question. **Fill ALL your (20) answers in the following table.** You should read the questions in the order given (as some questions are related).

Q1	6400		Q8	6		Q15	2
Q2	13		Q9	20.04		Q16	E (C)
Q3	630		Q10	3		Q17	D
Q4	690		Q11	2		Q18	1056
Q5	4		Q12	A		Q19	70
Q6	2		Q13	D		Q20	F
Q7	011		Q14	7			

1. Consider a relation R with 1600 pages, and we want to sort it using external merge sort. Assume that the DBMS uses quicksort for in-memory sorting (i.e., generate sorted runs of memory size in the first pass). Let B denote the number of buffers. What is the I/O cost to sort R given B is sufficient to sort R in two passes (i.e., generate run in one pass, and then merge these runs in a second pass)?

2. (2 marks) Consider the same setting in Question 1. What is the smallest number of buffers B that the DBMS can sort R using only three passes (i.e., generate sorted runs, and then need 2 more passes to merge the sorted runs)?

3. Consider relations R(a, b) and S(a, c, d) to be joined on the common attribute a.
   - There are B = 12 buffer pages
   - Table R comprises M = 200 pages with 80 tuples per page
   - Table S comprises N = 30 pages with 40 tuples per page

   What is the minimal cost (in terms of number of page accesses) to join R and S under Block Nested Loops Join? You should follow the approach taught in the lecture, i.e., you will need one buffer block to hold the evolving output block and one input block to hold the current input block of the inner relation. Also, ignore the cost of writing out the join results.

4. Consider the same setting in Question 3. What is the minimal cost (in terms of number of page accesses) to join R and S under GRACE Hash Join (as taught in the lecture)? Assume that the tuples are uniformly distributed across all partitions.

5. (2 marks) Insert the following 10 keys into an initially empty B+-tree of order 1:

   10, 30, 50, 70, 90, 14, 16, 20, 28, 24

   What is the height of the tree?

6. Consider the same setting in Question 5. Delete key 50 from the constructed B+-tree in Question 5. Consider the element/key 90. It appears X times in the tree. What is X?

7. Consider a linear hash table that is initially empty. After inserting some key values, the index now has buckets labelled from 000 to 101 (in binary). Which bucket (in binary) should a key with hash value 111 (binary) be sent to?

8. Consider a linear hash table whose buckets can hold up to 3 records. Suppose 18 records have been inserted into the hash table. Assume the splitting criterion as discussed in the lecture, i.e., split whenever there is an overflow. What is the

smallest number of buckets required (include overflow buckets, if necessary) to hold these records in the best case?

9. A certain disk unit has 10 surfaces, each with 100 tracks. Each track is divided into 1000 sectors, and a sector holds 2 KB. A page consists of 4 sectors. The disk rotates at 6000 rpm. The average seek time is 15ms. What is the time (in ms; up to 2 decimal points) to retrieve one random page? (NOTE: 1KB = 1024 bytes)

10. (2 marks) Consider the same disk in Question 9. Suppose we want to store relation R(A, B, C) on the disk as a contiguous sequential file ordered on the primary key attribute A. Suppose R has 1,000,000 fixed size records. The records in R are 200 bytes in size, and attribute A is 12 bytes (fixed size). Records do not span across sectors. We want to build a **sparse** B+-tree on A (one pointer per page of R). Suppose a pointer to a track requires 2 bytes. How many bytes (round up to the nearest byte) are required to represent a pointer to a page?

11. (2 marks) Consider the same setting as Question 10. Assume the B+-tree nodes are of the same size as a data page, and that all nodes are packed as full as possible. What is the height of the B+-tree?

12. It is generally better to organize a file cylinder by cylinder instead of surface by surface because organization by cylinder can
    A. reduce the head movement when reading the file.
    B. reduce the size of the bitmap for free space allocation.
    C. improve the rate at which data are transferred.
    D. allow a better performance during random access of a record.
    E. none of the above because organizing a file surface by surface is better!

13. Consider a system where most of the applications use small files. Which of the following is not a valid reason for avoiding large disk pages (e.g., one track)?
    A. Results in poor storage utilization
    B. Incurs long transfer time during disk access
    C. Requires a large I/O buffer to accommodate the data
    D. Lengthens the seek time during disk access
    E. None of the above

14. Consider the following sequence of numbers:

    22, 44, 7, 39, 49, 12, 89, 10, 66, 55, 50, 67, 40, 46

    Suppose we use replacement selection to generate sorted runs (as taught in lecture). Moreover, suppose each page has only 1 record. Let the number of buffer pages be 4. What is the length of the longest run?

15. Referring to Question 14. What is the number of sorted runs generated?

16. Consider the following three statements. Which of these are true?
   I) Sorting via a clustered index will always be better than applying the external multi-way sort algorithm.
   II) Sorting via an unclustered index will always be worse than applying the external multi-way sort algorithm.
   III) Sorting via a clustered index will not be inferior to sorting via an unclustered index.
   A) I only B) II only C) III only D) I and II only E) I and III only

17. Which of the following are reasonable criterion/criteria to be used for determining when to split a bucket in linear hashing?
   A. Whenever a bucket is full
   B. Whenever overall space utilization is 70%
   C. Whenever bucket pointed to by the "next" pointer overflows
   D. A & B E. B & C F. A & C G. A, B & C

18. Consider the join of two relations R and S. Suppose R and S are of the same size. What is the maximum size of R (and therefore S) for the two relations to be joined using GRACE hash join in two passes (as presented in the lecture) if we have a buffer pool of 34 pages? Assume the ideal case that every partition has the same number of tuples.

19. (2 marks) Consider the join of two relations R and S. Let |R| = |S| = 1000. Assume that the join can be performed in two passes using GRACE hash join (as in the lecture). Suppose we want to create the **minimum number of partitions** during the partitioning phase. Moreover, assume that we use one buffer for input, and the remaining buffers are split evenly for output partitions. In addition, for the joining phase, we **do not need** an output buffer as results are returned directly to users (without storing the output). Let the buffer size be 201 pages. What is the total **number of seeks** required to complete the join processing?

20. Which of the following statements is correct (include output cost)?
   A. The minimum cost to join two tables R and S (no restrictions on join algorithms, selectivities, memory size or output size, etc) is at least |R| + |S| where |R| and |S| denote the number of pages of R and S respectively.
   B. The cost to perform an intersection of two tables is the same as that to union the two tables.
   C. We can use an index-only approach to compute an arbitrary aggregate query if the attribute to be aggregated appears as part of the key in the index.
   D. A & C E. A & B F. None of A, B & C G. A, B & C

-- END OF PAPER –

Answers

1. 2 passes, so 2*1600*2 = 6400
2. For 3 passes. Generates runs of size B. Merge B-1 runs in each iteration. Since 2 iterations, so need (B-1)^2 runs. So need to make sure B*(B-1)*(B-1) is the smallest value larger than 1600. $13 * 12 * 12 = 1{,}872$. Anything less, fails.
3. Cost is minimal when S is the inner and R is the outer table. 3 iterations. 30 + 3*200 = 630
4. Cost is minimal when R is probing table. 3*(200+30) = 690
5. Build the tree. 4
6. Build the tree. 2
7. 011
8. 6
9. Time = avg seek time + avg rotational delay + transfer time. 15 + 10/2 + transfer 2 sectors. 10*4/1000 = 20.04 ms
10. 200 bytes/record. 20 records/sector. 40 records/page. 1000000/40 = 25,000 pages. 250 pages per track, so 8 bits are enough to represent a page within a track. So, need 1 bytes to represent a page within a track. To locate a page, we need its track number and then its page within the track, so we need in total 3 bytes to identify a page.
11. Each (key,RID)-pair takes 12+3 = 15 bytes. 2d*12 + (2d+1)*3 < 8096; d = 269. Each leaf node hold 538 (key, RID)-pairs. 25,000 pages requires 47 pages. So, 2 levels are enough (root + leaf).
12. A
13. D
14. 7
15. 2
16. E. I am accepting C as well (as the question is not very specific).
17. D
18. 1056 = 33 partitions of size 32
19. Partitioning phase: 5 partitions each 200 pages. Allocate 40 pages for each output partition. So, we need 5 seeks per partition. = (5 + 5*5) * 2. Joining Phase: build each partition takes 1 seek. So 5 for 5 partitions. Read S only needs 1 seek since we did not write out the output. Another 5 partitions. 5 seeks. = 5 + 5. Total = 70
20. F. (a) is wrong as it can be less than (|R|+|S|) (e.g., index-only scan or index join with very low selectivity). (b) is wrong because their output size will be different anyway. (c) is wrong because there may be a selection condition or projected attributes not in index.