Name: ______________________________ Student#: _________________________ Date: __________

COMP 3311: Database Management Systems

Final Review

1. Consider the relation Sailor(<u>sailorId</u>, name, rating, age), which is not sorted. Assume each attribute is 25 bytes, the page size is 1,000 bytes and there are 11,000 tuples. For the following questions, apply external sorting using a buffer of 11 pages.

   A. How many sorted runs will be produced in pass 0?

      a) 10
      b) 11
      c) 100
      d) 110
      e) None of the above.

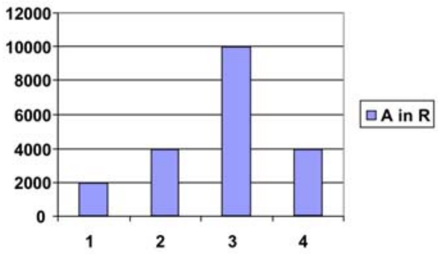
   B. What is the <u>total number of passes</u> required to sort the result completely (including pass 0)?

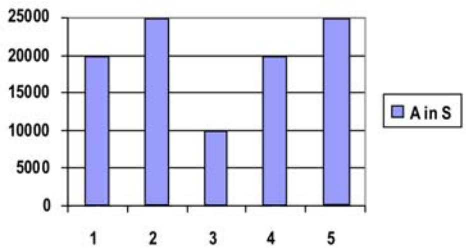
      a) 1
      b) 2
      c) 3
      d) 4
      e) 5

   C. What is the total page I/O cost of sorting?

      1. 4,400
      2. 5,500
      3. 6,600
      4. 7,770
      5. None of the above.

2. Consider the two relations R(A, $\underline{B, C}$) and S(A, $\underline{B}$, Y). R contains 20,000 tuples and S contains 100,000 tuples. Assume that for both relations 10 tuples fit per page (i.e., the size of R is 2,000 pages and that of S is 10,000 pages). The possible values of attribute A in R are {1, 2, 3, 4}, whereas the possible values of A in S are {1, 2, 3, 4, 5}. The following histograms present statistical information about the occurrences of values for A in R and S (e.g., there are 2,000 tuples with A=1 in R and 20,000 tuples with A=1 in S).





A. How many tuples are there in the result of the query (R $JOIN_{R.A=S.A}$ S) (i.e., what is the cardinality of the output)?

   a) 20,000
   b) 100,000
   c) 120,000
   d) 320,000
   e) 320,000,000

B. What is the minimum cost (in terms of page I/Os) for computing (R $JOIN_{R.A=S.A}$ S) using block nested-loop join and how many main memory pages are needed?

   a) Minimum cost is 12,000 and I need 2,000 main memory pages.
   b) Minimum cost is 12,000 and I need 2,002 main memory pages.
   c) Minimum cost is 12,000 and I need 12,000 main memory pages.
   d) Minimum cost is 320,000 and I need 2,002 main memory pages.
   e) Minimum cost is 320,000 and I need 12,000 main memory pages.

C. We want to compute (R $JOIN_{R.A=S.A}$ S) using block nested-loop join with *R as the outer relation*. What is the minimum number of main memory pages needed in order to achieve a cost of 42,000 page I/Os?

   a) 502
   b) 736
   c) 1,000
   d) 1,002
   e) 2,000

D. We want to compute (R $JOIN_{R.A=S.A}$ S) using hash join and *R as the build input*. How many buckets should be used for partitioning and what is the minimum main memory requirement?

   a) 4 buckets and at least 202 main memory pages.
   b) 4 buckets and at least 1,002 main memory pages.
   c) 10 buckets and at least 202 main memory pages.
   d) 10 buckets and at least 502 main memory pages.
   e) 20 buckets and at least 102 main memory pages.

E. Given that R.B is a NOT NULL foreign key referencing S.B, how many tuples are in the result of the query (R $\text{JOIN}_{R.B=S.B}$ S)?

a) 20,000
b) 100,000
c) 120,000
d) 320,000
e) 320,000,000

F. We want to compute (R $\text{JOIN}_{R.B=S.B}$ S) using indexed nested-loop join with *R as the outer relation*. Assume that there is a hash index on S.B with no overflow buckets (i.e., finding an index entry has cost 1). What is the total I/O cost of the join?

a) 6,000
b) 22,000
c) 40,000
d) 42,000
e) None of the above.

G. How many tuples are there in the result of (($\sigma_{A=1}$R) $\text{JOIN}_{R.B=S.B}$ S) and what is the minimum cost of processing the query using indexed nested-loop join with *R as the outer relation*. Assume that the only index is a hash index on S.B with no overflow buckets.

a) The result has 2,000 tuples and the cost is 6,000 page I/Os.
b) The result has 2,000 tuples and the cost is 22,000 page I/Os.
c) The result has 20,000 tuples and the cost is 40,000 page I/Os.
d) he result has 20,000 tuples and the cost is 42,000 page I/Os.
e) None of the above.

H. How many tuples are expected in the result of (($\sigma_{A=1}$R) $\text{JOIN}_{R.B=S.B}$ ($\sigma_{A=3}$S)) and what is the minimum cost of processing the query using indexed nested-loop join with *R as the outer relation*. Assume that the only index is a hash index on S.B with no overflow buckets.

a) Expected number of tuples is 200 with cost 600 page I/Os.
b) Expected number of tuples is 200 with cost 6,000 page I/Os.
c) Expected number of tuples is 2,000 with cost 6,000 page I/Os.
d) Expected number of tuples is 2,000 with cost 42,000 page I/Os.
e) None of the above.

3. Consider the following schedules of two transactions T_1 and T_2. Indicate for each whether it is serial, (conflict) serializable or not serializable. r denotes a READ and w a WRITE operation.

a) Schedule: $r_1(A)$ $w_1(A)$ $r_2(A)$ $w_2(B)$

b) Schedule: $r_1(A)$ $r_2(A)$ $w_1(A)$ $w_2(B)$

c) Schedule: $r_1(A)$ $r_2(A)$ $w_2(A)$ $w_1(A)$

4. Consider the schedule $r_1(A)$ $w_1(A)$ $r_2(A)$ $w_2(B)$ c_1 c_2 (where c_1 and c_2 indicate the commit statements).

   a) Is the schedule recoverable and cascadeless? Why?

   b) Change the time of the commits (c_1, c_2) in the schedule in a) so that it becomes a cascadeless schedule.

   c) Is the schedule $r_2(A)$ $r_1(A)$ $w_1(A)$ $w_2(B)$ c_2 c_1 recoverable and cascadeless?

5. Rewrite the schedule $r_2(A)$ $r_1(A)$ $w_1(A)$ $w_2(B)$ according to the 2PL protocol (i.e., add lock-S, lock-X, unlock statements below). Explain briefly whether the schedule is allowed by 2PL.

T_1	T_2
	read(A)
read(A)	
write(A)	
	write(B)

6. Rewrite the schedule $r_2(A)$ $r_1(A)$ $w_1(A)$ $w_2(B)$ according to timestamp-ordering protocol (i.e., add RTS (read timestamp) and WTS (write timestamp) statements). Assume that the timestamps of T_1 and T_2 are 2 and 1, respectively. The initial read and write timestamps of A and B are both 0.

T_1 [TS=2]	T_2 [TS=1]
	read(A)
read(A)	
write(A)	
	write(B)

7. Rewrite the schedule $r_2(A)$ $r_1(A)$ $w_1(A)$ $w_2(B)$ according to the multi-version timestamp-ordering protocol (i.e., add RTS (read timestamp) and WTS (write timestamp) statements and specify the versions of the items). Assume that the timestamps of T_1 and T_2 are 1 and 2, respectively and that the initial versions of items are A_0 and B_0. Complete the correct version numbers (e.g., read(A_0) instead of read(A)).

T_1 [TS=1]	T_2 [TS=2]
	read(A)
read(A)	
write(A)	
	write(B)

You must upload this completed exercise sheet to Canvas by **11:00 p.m. today.**