CSE 232A Fall 2018 Midterm Exam 1

Full Name:

Student ID:

INSTRUCTIONS

1. You have 50 minutes to complete this exam.

2. You can have up to one letter/A4-sized sheet of notes, formulae, etc. Apart from this, the exam is closed book/notes/electronics/peers.

3. Please wait until being told to start reading and working on the exam.

4. If you think a question is ambiguous, write down your assumptions, argue that they are reasonable, and then work on the problem using those assumptions.

5. Please ensure that your writing is clear and legible!

<table>
<thead>
<tr>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q 1</td>
<td>7</td>
</tr>
<tr>
<td>Q 2</td>
<td>11</td>
</tr>
<tr>
<td>Q 3</td>
<td>9</td>
</tr>
<tr>
<td>Q 4</td>
<td>12</td>
</tr>
<tr>
<td>Q 5</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
</tr>
</tbody>
</table>
Hi, this is your son's school. We're having some computer trouble.

Oh, dear - did he break something? In a way...

Did you really name your son Robert? DROP TABLE Students;--?

Oh, yes. Little Bobby Tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.
Q 1. [7pts] For the following questions, clearly circle True or False.

1. Sequential data access is typically orders of magnitude faster than random data access when reading data from non-volatile memory.

   True  False

2. When the pin count of a buffer frame is decremented, the page in that frame is said to be “unpinned.”

   True  False

3. In the unpacked layout for fixed-length records, deleting a record will alter the record IDs of other records on the same page.

   True  False

4. In the delimiter-based record format, we need to scan the record from the start even to retrieve a single field in the middle.

   True  False

5. It is possible to have multiple clustered B+ tree indexes on the same table.

   True  False

6. Redistribution of index keys among siblings following the overflow of a non-leaf node is the first preference for reorganizing a B+ tree index after an insert operation.

   True  False

7. A key benefit of a learned index over a B+ tree index reported in Kraska et al. (SIGMOD 2018) is that the former can be much smaller in size.

   True  False
Q 2. [11pts] Consider the following extendible hash index with 4 slots per bucket.

1. [5pts] What is the global depth after the following sequence of operations: delete 32*, insert 32*, insert 125*, insert 17*, and insert 36*?

   (a) 1  (b) 2  (c) 3  (d) 4  (e) 5  (f) 6

2. [2pts] After the above operations, what is the largest number of directory pointers pointing to the same bucket?

   (a) 1  (b) 2  (c) 3  (d) 4  (e) 5  (f) 6

3. [4pts] After the above operations, what is the minimum number of delete operations needed now for the global depth to decrease?

   (a) 1  (b) 2  (c) 3  (d) 4  (e) 5  (f) 6
Q 3. [9pts] Suppose we are sorting a relation with 1 million pages and we have 100 buffer pages for the external merge sort (EMS). A "pass" over the relation is defined as one read and write of the whole file. In all of the following, you have to include both the sort and merge phases. **Clearly circle** the correct answer for each of the following questions.

1. [3pts] How many passes will a multi-way EMS perform, assuming we use replacement sort for internal sorting?

   (a) 2  (b) 3  (c) 4  (d) 5  (e) 6  (f) 7

2. [3pts] How many passes will a multi-way EMS perform, assuming we do not use replacement sort for internal sorting but use blocked I/O with block sizes of 10 pages?

   (a) 2  (b) 3  (c) 4  (d) 5  (e) 6  (f) 7

3. [3pts] How many passes will a multi-way EMS perform, assuming we use replacement sort for internal sorting along and double buffering but no blocked I/O?

   (a) 2  (b) 3  (c) 4  (d) 5  (e) 6  (f) 7
Q 4. [12pts] We are given two tables $R(X, Y)$ and $S(X, Y)$ with the same schema, i.e., they are union-compatible. Assume attributes $X$ and $Y$ are attributes of the same size. The table sizes of $R$ and $S$ happen to be $1.8B$ and $6B$ pages respectively, where $B$ is the number of buffer pages given (in the millions). Assume there are no skews in any attribute distributions. Hash table fudge factor is 1.4.

Given all the above, what is the I/O cost for executing each of the following queries in terms of $B$ using the specified physical operator implementations? Exclude the cost of writing the output.

1. [3pts] $\pi_X(R)$ with sorting-based project
   
   (a) $1.8B$  (b) $2.7B$  (c) $3B$  (d) $3.6B$  (e) $4.5B$

2. [3pts] $\pi_Y(S)$ with hashing-based project
   
   (a) $3B$  (b) $6B$  (c) $9B$  (d) $12B$  (e) $15B$

3. [6pts] $R \bowtie S$ with block-nested loop join (in-memory hash table is used on blocks)
   
   (a) $11.4B$  (b) $19.8B$  (c) $22.2B$  (d) $23.4B$  (e) $31.6B$
Q 5. [10pts] Are you on social media? Given the following relational database schema, translate each SQL query over this relational database into an equivalent relational algebra query. Circle the correct answer (only one is correct).

Person (ID, Name, Age)
Friends (ID1, ID2)

Person.ID is the primary key of Person. Both Friends.ID1 and Friends.ID2 are foreign keys referring to Person.ID.

1. [2pts] SELECT DISTINCT Name FROM Person WHERE Age > 20

(a) \( \pi_{\text{Name}}(\sigma_{\text{Age} > 20}(\text{Person})) \)
(b) \( \sigma_{\text{Age} > 20}(\pi_{\text{Name}}(\text{Person})) \)
(c) \( \pi_{\text{Name,ID}}(\sigma_{\text{Age} > 20}(\text{Person})) \)
(d) \( \sigma_{\text{Age} > 20}(\pi_{\text{Name,ID}}(\text{Person})) \)
(e) \( \sigma_{\text{Name}}(\pi_{\text{Age} > 20}(\text{Person})) \)
(f) \( \pi_{\text{Name}}(\pi_{\text{Age}}(\sigma_{\text{Age} > 20}(\text{Person}))) \)

2. [2pts] SELECT ID1, COUNT(*) FROM Friends GROUP BY ID1

(a) \( \sigma_{\text{ID1}}(\gamma_{\text{ID1,COUNT(*)}}(\text{Friends})) \)
(b) \( \pi_{\text{ID1}}(\gamma_{\text{COUNT(*)}}(\text{Friends})) \)
(c) \( \gamma_{\text{ID1,COUNT(*)}}(\sigma_{\text{ID1,ID2}}(\text{Friends})) \)
(d) \( \sigma_{\text{ID1,COUNT(*)}}(\text{Friends}) \)
(e) \( \pi_{\text{ID1,COUNT(*)}}(\text{Friends}) \)
(f) \( \gamma_{\text{ID1,COUNT(*)}}(\text{Friends}) \)
3. [6pts] SELECT DISTINCT P2.Name
FROM Person P1, Person P2, Friends F
WHERE P1.ID = F.ID1 AND P2.ID = F.ID2 AND P1.Name = "Thanos"

(a) \( \pi_{Name}(\pi_{ID1,ID2}(\pi_{ID}(\sigma_{Name="Thanos"}(Person))) \bowtie_{ID=ID2} Friends) \bowtie_{ID2=ID} Person) \)

(b) \( \pi_{Name}(\pi_{ID1,ID2}(\pi_{ID}(\sigma_{Name="Thanos"}(Person))) \bowtie_{ID=ID1} Friends) \bowtie Person) \)

(c) \( \pi_{Name}(\pi_{ID1,ID2}(\pi_{ID}(\sigma_{Name="Thanos"}(Person))) \bowtie_{ID=ID1} Friends) \bowtie_{ID2=ID} Person) \)

(d) \( \pi_{Name}(\pi_{ID1}(\pi_{ID}(\sigma_{Name="Thanos"}(Person))) \bowtie_{ID=ID1} Friends) \bowtie_{ID2=ID} Person) \)