CS W186 Spring 2019 Midterm 2

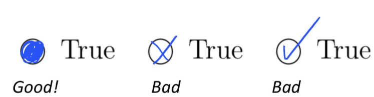
Do not turn this page until instructed to start the exam.

Contents:

- You should receive one *double-sided answer sheet* and a 11-page *exam packet*.
- The midterm has *6 questions*, each with multiple parts.
- The midterm is worth a total of *76 points*.

Taking the exam:

- You have *110 minutes* to complete the midterm.
- All answers should be written on the answer sheet. The exam packet will be collected but not graded.
- For each question, place only your *final answer* on the answer sheet; do not show work.
- For multiple choice questions, please *fill in the bubble or box completely* as shown on the left below. *Do not mark the box with an X or checkmark.*



- Use the blank spaces in your exam for scratch paper.

Aids:

- You are allowed **two handwritten** 8.5" × 11" double-sided pages of notes.
- The **only** electronic devices allowed are basic scientific calculators with simple numeric readout. No graphing calculators, tablets, cellphones, smartwatches, laptops, etc.

Grading Notes:

- All IOs must be written as integers. There is no such thing as 1.04 IOs – that is actually 2 IOs.
- 1 KB = 1024 bytes. We will be using powers of 2, not powers of 10
- Unsimplified answers, like those left in log format where simplification to integers is possible, will receive a point penalty.

1 Joins (18 points)

Consider the following relations

```
CREATE TABLE Customers (
      cid INTEGER PRIMARY KEY,
      name CHAR(30),
      address CHAR(30)
);

CREATE TABLE Orders (
      oid INTEGER PRIMARY KEY,
      cid INTEGER REFERENCES Customers(cid),
      item_name CHAR(20),
      item_count INTEGER
);
```

In this problem, we will consider executing the following query using various join algorithms.

```
SELECT *
FROM Customers C, Orders O
WHERE C.cid = O.cid;
```

Assume that

- `Orders` has $[O] = 100$ pages, with $p_O = 50$ tuples per page.
- `Customers` has $[C] = 40$ pages, with $p_C = 25$ tuples per page.
- We have an Alternative 2 B+ tree index on `Orders.cid` with height $h = 2$ (recall that a tree with only the root has height 0).
- Each customer has at least one order, and `Orders.cid` is uniformly distributed.
- Our buffer has size $B = 10$, unless noted otherwise.
- For index nested loops join, assume the cost model from lecture, where we do not cache index pages.
- Unless otherwise noted, each of the following parts are to be answered independently, i.e. assume the query is executed from scratch for each part.
- The files are not already sorted.

1. (2 points) In the best case, what is the I/O cost of a block nested loops join?
2. (2 points) Assume the index on `Orders.cid` is clustered. In the best case, what is the I/O cost of an index nested loops join?
3. (2 points) Assume the index on `Orders.cid` is unclustered. In the worst case, what is the I/O cost of an index nested loops join?
4. (2 points) In the best case, what is the I/O cost of a (unoptimized) sort-merge join?
5. (3 points) What is the minimum buffer size B in order to perform the optimized sort-merge join? Recall that the optimized sort-merge join uses 2 passes: 1 sorting pass, and 1 pass for both merging and joining.
6. (2 points) Assuming that our buffer is big enough, what is the I/O cost of the optimized sort-merge join described in the previous part?

7. (1 point) Assume we do a grace hash join and that `Customers` is the building relation, i.e. we build the in-memory hash table using `Customers`. Assume after the first partitioning pass, we partition `Customers` into the following partitions (assume all pages are full):

- P_{C1} : 10 pages
- P_{C2} : 9 pages
- P_{C3} : 9 pages
- P_{C4} : 8 pages
- P_{C5} : 4 pages

with the remaining partitions having 0 pages. What is the size (in pages) after the first partitioning pass of the partitions $P_{O1}, P_{O2}, P_{O3}, P_{O4}, P_{O5}$ respectively (as a tuple of five integers)? Here P_{Oi} denotes the i-th partition of `Orders` after the first partitioning pass, assuming the same ordering of partitions as the P_{Ci} partitions described above.

   A. (50, 45, 45, 40, 20)

   B. (25, 22, 22, 20, 10)

   C. (12, 12, 12, 12, 12)

   D. (25, 23, 23, 20, 10)

   E. (20, 20, 20, 20, 20)

8. (4 points) Assume that the hash function used for recursive partitioning h_r partitions each partition of `Customers` uniformly (or as close to uniformly as possible) among 2 new partitions. What is the I/O cost of grace hash join from the previous part? Recall that we only perform recursive partitioning when necessary, i.e. we further divide only the partitions that are too large for the build and probe phase.

2 Parallelism (12 points)

For this question, assume the following:

- Assume that we have three tables: `Teachers(name, salary, course)`, `Students(name, SID, course, grade)`, `Employees(name, eid, company, salary)`
- $[T] = 10$ pages
- $[S] = 10000$ pages
- $[E] = 100000$ pages
- 1 page = 4 KB
- We have 4 machines with 52 buffer pages each
- Assume each I/O takes 1 ms. There are 1000ms in a second.
- All questions are to be considered independently of each other.

1. (2 points) Assume `Students` is round-robin partitioned across the 4 machines, and that `Teachers` lies entirely on Machine 1. What is the minimum network cost **(in KB)** incurred from performing a join between `Teachers` and `Students`?
2. (2 points) If `Students` is round-robin partitioned across the 4 machines and `Employees` is range-partitioned on the `name` key across 4 machines, what would be the worst-case network cost be **(in KB)** to perform a sort-merge join between `Students` and `Employees` on the `name` key?
3. (2 points) Let us assume `Students` and `Employees` are both hash-partitioned on the `name` key using the same hash function across the 4 machines and that, due to key skew, machine 1 has 50% of `Students` on it, with the remaining 50% of the Students table distributed evenly among machines 2, 3, and 4. Assume `Employees` is distributed uniformly across all 4 machines. If we were to execute a block nested loop join between `Students` and `Employees` on each machine, how much time would be taken, in seconds, to complete this? Ignore the time it takes to write the output.
4. (2 points) Assuming Students is range-partitioned across the 4 machines with 40% of Students going to machine 1 and the remaining 60% split evenly across machines 2, 3, and 4. How long, in seconds, would it take to parallel sort the Students table?
5. (2 points) Now, assume `Students` is range-partitioned on the `name` key evenly across the 4 machines. How long, in seconds, would it take to parallel sort the `Students` table on this key?
6. (2 points) Now, we wish to calculate the square root of the sum of the squared salary of each row in `Employees`, i.e. `SELECT SQRT(SUM(salary * salary)) FROM Employees`. Mark a global and local aggregate function that would allow us to efficiently calculate `SQRT(SUM(salary * salary))` via hierarchical aggregation.

Local	Global
A. count(x)	A. count(x)
B. x	B. x
C. $\sum x$	C. $\sum x$
D. $\sum x^2$	D. $\sum x^2$
E. $\sqrt{\sum x}$	E. $\sqrt{\sum x}$
F. $\sqrt{\sum x^2}$	F. $\sqrt{\sum x^2}$

3 Query Optimization (15 points)

For questions 1-4, we consider the following schema:

```
CREATE TABLE Guitars(
    gid INTEGER PRIMARY KEY,
    brand VARCHAR(50),
    price INTEGER
);
CREATE TABLE Players(
    pid INTEGER PRIMARY KEY,
    name VARCHAR(50),
    age INTEGER
);
CREATE TABLE LastPlayed(
    gid INTEGER REFERENCES Guitars(gid),
    pid INTEGER REFERENCES Players(pid),
    date DATE,
    PRIMARY KEY (gid, pid)
);
```

We make the following assumptions about the distribution of data:

- $10 \leq$ `Players.age` < 85
- $1{,}000 \leq$ `Guitars.price` $< 5{,}000$
- `Players.pid` has 1,000 distinct values
- `Guitars.gid` has 1,000 distinct values
- `Guitars.brand` has 10 distinct values
- `Players.age` is independent from `Guitars.brand`

Consider the following query:

```
SELECT P.name
  FROM Guitars G, Players P, LastPlayed L
 WHERE G.gid = L.gid AND P.pid = L.pid
   AND (P.age < 25 OR G.brand = 'CS186')
   AND G.price >= 3000;
```

Compute the selectivity for each of the following predicates from the `WHERE` clause. Write only your final answer, as a simple fraction.

1. (1 point) `G.gid = L.gid`

2. (1 point) `P.pid = L.pid`

3. (1 point) `G.price >= 3000`

4. (1 point) `P.age < 25 OR G.brand = 'CS186'`

For the next question, consider the following schema:

```
CREATE TABLE Product(
    pid INTEGER PRIMARY KEY,
    name TEXT,
    price INTEGER
);
CREATE TABLE Company(
    cid INTEGER,
    pid INTEGER REFERENCES Product(pid),
    name TEXT,
    PRIMARY KEY (cid, pid)
);
```

- Product contains 20,000 tuples, and each record is 20 bytes long.
- Company contains 1,000 tuples, and each record is 25 bytes long.
- Each page can hold 5,000 bytes.
- The buffer pool is 102 pages large.
- The fill factor for all hash tables is 0.8.
- There are no indices.

5. (2 points) Consider all the join algorithms covered so far in this class. What is the minimum estimated I/O cost to execute the following query? *Exclude the final write from your solution.*

```
SELECT P.name, C.name
  FROM Product P, Company C
 WHERE P.pid = C.pid
```

For the next two questions, we consider the following query on the relations `R(a,b,c,d,e)`, `S(a,b,c,d,f)`, `T(a,b,c,d)`, `U(a,b,c,d)`:

```
SELECT * FROM R
   INNER JOIN S ON R.a = S.a AND R.b = S.b
   INNER JOIN T ON S.c = T.c AND S.d = T.d
   INNER JOIN U ON U.c = R.c AND R.a = U.a
 WHERE R.e > 1000 AND S.f > 0
 GROUP BY S.c
 ORDER BY R.b LIMIT 10;
```

6. (2 points) For each of the following joins, mark True if the Selinger optimizer considers it at some point, and False if not.

   A. $(R \bowtie S) \bowtie T$

   B. $(R \bowtie U) \bowtie (S \bowtie T)$

   C. $((S \bowtie U) \bowtie R) \bowtie T$

   D. $((R \bowtie U) \bowtie S) \bowtie T$

7. (4.5 points) Consider the following table while running a pass of the Selinger optimizer's dynamic programming algorithm:

	Left Relations	Right Relation	Sorted on	I/O cost	Output Size
(a)	{S, T, U}	R	N/A	10,000	4,000
(b)			R.a	12,000	
(c)			(R.b, R.a)	11,000	
(d)	{R, T, S}	U	N/A	2,000	4,000
(e)			(R.a, R.b)	52,000	
(f)	{R, U, T}	S	N/A	22,000	4,000
(g)			S.c	100,000	
(h)	{R, U, S}	T	N/A	150,000	4,000
(i)			S.c	110,000	

   For each row, mark whether the row is retained at the end of the pass.

For each of the following questions, mark either True or False.

8. (0.5 points) The Selinger algorithm will produce an optimal query plan, given a perfect cost estimator.

9. (0.5 points) An output is sorted on column `T.c`, where `T.c` is used in a join in the query. Then, we have an interesting order on `T.c`.

10. (0.5 points) If we only consider BNLJ and do not consider interesting orders while joining n tables, the number of left-deep query plans is $O(n!)$

11. (0.5 points) True or False: the number of joins considered over the course of the dynamic programming algorithm is always at least exponential in the number of tables we are selecting from.

12. (0.5 points) Consider a relation `R` which is sorted on some column `c` and `R.c` is used in an ORDER BY clause. A page nested loop join with `R` as the outer relation produces an interesting order of `R.c`.

4 Text Search (8 points)

1. (2 points) For each assertion, fill in the corresponding bubble True or False.
   A. In the "Bag of Words" model, we might choose to ignore the word "a" in phrase "a table" because it doesn't contain much information. This is an example of a stop word.
   B. In vector space model, the dimension of vectors depends on the length of the longest document.
   C. In text search engine, querying is efficient but updates are not.
   D. The IDF part of the TF-IDF value is responsible for favoring repeated terms.

2. (2 points) We have 64 documents, and the following table summarizes the number of documents containing a term and the number of occurrences of a term in `doc1` for several terms:

term	# docs containing term	# terms in `doc1`
Algorithm	4	20
Berkeley	16	17
Computer	1	12
Database	16	21

   What is the TF-IDF for term = "Database" for `doc1`?. *Assume base-2 for any logarithms.*

3. (2 points) We are given documents d1 and d2 with a "bag of words" containing the vocabulary (Dog, Kangaroo, Cat, String, Scratch, Wood). We decide to model them with the following vectors: $d_1 = [1, 3, 2, 1, 1, 0]^\top$ and $d_2 = [1, 0, 0, 0, 2, 2]^\top$. Find the cosine similarity between d_1 and d_2. Write your answer as a decimal, rounded to two decimal places.

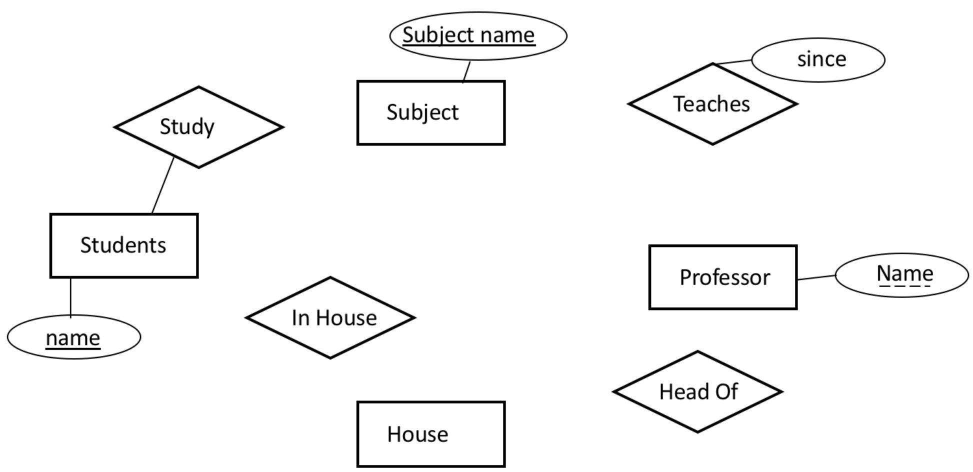
4. (1 point) Suppose that there exist 186 documents about database systems in our corpus. We query for documents about database systems, and our query returns 25 results, 13 of which are actually about database systems. What is the precision? Express your answer as a fraction, you do not need to simplify.

5. (1 point) What is the recall from the previous part (expressed as a fraction)?

5 ER Diagrams (11 points)

Welcome to Hogwarts! As a new student you are trying to understand how the school works. You are given the diagram below, but some information is missing! Fill in the diagram using the details below:

- A subject must have at least one student studying it
- A professor is uniquely identified by their name and the subject they teach
- Every subject must have at least one professor teaching it
- Each house has exactly one professor who is the head of house.
- Each professor can be the head of at most one house
- Every house must have at least one member
- Every student must be in exactly one house



1. (1 point) What type of edge should be drawn between the **Students** entity and the **In House** relationship set?
   - A. Thin Line
   - B. Bold Line
   - C. Thin Arrow
   - D. Bold Arrow
2. (1 point) What type of edge should be drawn between the **House** entity and the **In House** relationship set?
   - A. Thin Line
   - B. Bold Line
   - C. Thin Arrow
   - D. Bold Arrow
3. (1 point) What type of edge should be drawn between the **Professor** entity and the **Teaches** relationship set?

A. Thin Line
B. Bold Line
C. Thin Arrow
D. Bold Arrow

4. (1 point) What type of edge should be drawn between the **Subject** entity and the **Study** relationship set?
    A. Thin Line
    B. Bold Line
    C. Thin Arrow
    D. Bold Arrow

5. (1 point) What type of relation is **Professor** to **Head of**?
    A. many-to-many
    B. many-to-one
    C. one-to-many
    D. one-to-one

6. (1 point) What type of edge should be drawn between the **Professor** entity and the **Head of** relationship set?
    A. Thin Line
    B. Bold Line
    C. Thin Arrow
    D. Bold Arrow

7. (1 point) True or False: **Students** are required to study one or more **Subjects**.
    A. True
    B. False

8. (1 point) True or False: Multiple **Professors** may teach one **Subject**
    A. True
    B. False

9. (1 point) True or False: The **Professor** entity is a weak entity
    A. True
    B. False

10. (2 points) In their fifth year, students may be nominated as prefect. Each house may have one female and one male prefect. Which of the following would be the effective way to represent this in the diagram?
    A. A new relationship set between Students and House called Prefect with a bold line connecting Student to Prefect and a thin line connecting House to Prefect because it is a many-to-one relationship
    B. A new relationship set between Students and House called Prefect with a bold arrow from Student to Prefect and 2 bold arrows from House to Prefect
    C. 2 new relationships between Students and House called Female Prefect and Male Prefect with a thin arrow from Student to Female Prefect and a bold arrow from House to Female Prefect. Also a thin arrow from Student to Male Prefect and a bold arrow from House to Male Prefect.
    D. This is impossible. You cannot have this type of relationship in an ER diagram

6 Functional Dependencies (12 points)

1. (5 points) Decompose R = ABCDEF into BCNF in the order of the following functional dependencies: AE $\rightarrow$ F, A $\rightarrow$ B, BC $\rightarrow$ D, CD $\rightarrow$ A, CE $\rightarrow$ D.

   Which of the following tables are included in the final decomposition?

   A. ABC

   B. ACD

   C. AEF

   D. BCD

   E. CDE

For the next 4 questions, consider relation R = ABCDEF and functional dependencies F = {AB $\rightarrow$ CF, CE $\rightarrow$ B, F $\rightarrow$ D}.

2. (2.5 points) Which of the following are superkeys of this relation?

   A. AB

   B. ABE

   C. ACE

   D. ABCD

   E. ABCDE

3. (2.5 points) Which of the following are candidate keys of this relation?

   A. AB

   B. ABE

   C. ACE

   D. ABCD

   E. ABCDE

4. (1 point) True or False: The decomposition of R into ABCE and ABDEF is a dependency preserving decomposition.

5. (1 point) True or False: The decomposition of R into ABCE and ABDEF is a lossless decomposition.