CSE 544: Homework 4

Due: Monday, June 10, 2013, 11:59pm

Name: _____

Question	Points	Score
1	60	
2	60	
3	30	
4	30	
5	20	
Total:	200	

You may turn in this homework either electronically (in pdf or word format), or print a copy of the assignment, write your answers in the spaces provided, then turn it in to Kevin.

1 Datalog

1. (60 points)

   (a) (10 points) Consider the following two datalog programs computing the transitive closure:

```
P1:
    T(x,y) :- R(x,y)
    T(x,y) :- T(x,z),R(z,y)

P2:
    T(x,y) :- R(x,y)
    T(x,y) :- T(x,z),T(z,y)
```

   Suppose R is a graph that consists of the following n edges:

$$R(a_0, a_1), R(a_1, a_2), \ldots, R(a_{n-1}, a_n)$$

   Assume that we evaluate both programs using the semi-naive evaluation algorithm.

   i. How many facts does the IDB predicate T contain at the end of the program?

   i. $\binom{n}{2}$

   Your answer:

   Solution: T consists of all $\binom{n}{2}$ ground facts for the form $T(a_i, a_j)$, for $1 \leq i < j \leq n$.

   ii. For a fixed $m = 1, \ldots n-1$, how many times will the fact $T(a_1, a_{m+1})$ be discovered by `P1`?

   ii. **One time**

   Your answer:

   Solution: Once, because $T(a_1, a_{m+1})$ can be uniquely written as $T(a_1, a_m), R(a_m, a_{m+1})$.

   iii. How many times will the fact $T(a_1, a_{m+1})$ be discovered by `P2`?

   iii. $m-1$

   Your answer:

Solution: There are $m-1$ ways to discover $T(a_1, a_{m+1})$:

$$\begin{aligned} T(a_1, a_{m+1}) =& T(a_1, a_2), T(a_2, a_{m+1}) \\ T(a_1, a_{m+1}) =& T(a_1, a_3), T(a_3, a_{m+1}) \\ & \dots \\ T(a_1, a_{m+1}) =& T(a_1, a_m), T(a_m, a_{m+1}) \end{aligned}$$

(b) (10 points) Consider the following datalog program that finds the set of nodes accessible from those in a collection `Source`:

```
P1:
  T(x,y) :- R(x,y
  T(x,y) :- T(x,z),R(z,y)
  A(y)   :- Source(x),T(x,y)
```

A much more efficient way to compute the predicate $A(x)$ is by the following datalog program:

```
P1':
  A(y) :- Source(x), R(x,y)
  A(y) :- A(x),R(x,y)
```

i. Suppose R is a graph with n nodes. What is the maximum number of IDB facts computed by P1? Your answer should include IDB facts for both IDB predicates, T and A.

i. $n^2 + n$

Your answer:

ii. Same question for P1': what is the maximum number of IDB facts computed by P1'? Notice that the only IDB predicate is A.

ii. $n^2 + n$

Your answer:

iii. The transformation from `P1` to `P1'` is called *magic set optimization*. Now consider the following datalog program computing all nodes in the same generation as those in `Source`:

```
P2:
  Sg(u,v) :- R(x,u),R(x,v)
  Sg(u,v) :- Sg(x,y),R(x,u),R(y,v)
  A(v)    :- Source(u), Sg(u,v)
```

Write a different program `P2'` that computes the same answer A as `P2` and is more efficient, in that it computes fewer facts than `P1`. Your solution `P2'` should be a best-effort solution. You may find it useful to think about the case when R is a binary tree and design your program for this case, but your solution should work in general, on any graph. Hint: you may google for magic sets optimizations, but you may find magic sets quite confusing, so it may be easier if you come up with `P2'` yourself.

Solution:

```
P2':
  B(x) :- R(x,u),Source(u)    // find the ancestors of all Source nodes
  B(x) :- R(x,u),B(u)  //
  Sb(u,v) :- B(u),R(x,u),R(x,v)     // compute the same generation only for
  Sb(u,v) :- B(u),Sb(x,y),R(x,u),R(y,v)
  A(v) :- Source(u),Sb(u,v)  // same as above
```

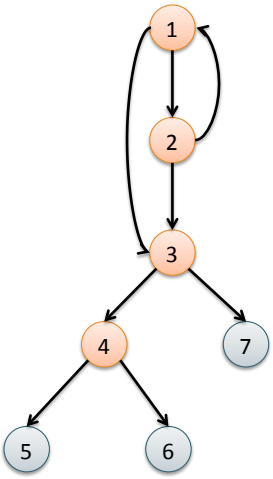

Figure 1: Example of the Win-Move game. Alice has a winning strategy if the game starts at nodes 3, 5, 6, or 7. Bob has a winning strategy if the game starts at node 4, because Alice makes the first move and she is forced to move to a leaf node. Nobody has a winning strategy if the game starts at nodes 1 or 2.

(c) (10 points) The following game is called the *Win-Move* game. The game is played on a graph T where each node x is either a leaf node, or has two outgoing edges $(x, y), (x, z)$. All leaf nodes x are stored in a relation $L(x)$; all non-leaf nodes x with children y, z are stored in a relation $T(x, y, z)$.

The Win-Move game is the following. There are two players, Alice and Bob. The players move a pebble on the graph: Alice makes the first move, then they take turns. If the pebble is on a non-leaf node x, then the player whose turn it is moves it to one of the two children, y or z. If the pebble is on a leaf node, then the player picks up the pebble and wins the game.

Write a datalog program to computes the set nodes where Alice has a guaranteed winning strategy. Your program should compute a relation `Alice`(x) that returns all nodes x such that, if the game starts with the pebble on x and Alice is smart, then we can win no matter what Bob does. For example, consider the graph in Figure 1. Node 3 is a winning node for Alice: if the game starts with the pebble on node 3, then Alice will move it to 4, and no matter where Bob moves it, Alice wins at the next round. However, Node 1 has no a winning strategy for Alice: if she moves to node 4 then Bob will win, and if she moves to 2 then Bob will move back to 1, resulting in an infinite game.

Solution:

```
Alice(x) :- L(x)
Alice(x) :- T(x,y,z),Bob(y)
Alice(x) :- T(x,y,z),Bob(z)
Bob(x)   :- T(x,y,z),Alice(y),Alice(z)
```

(d) (10 points) Prove that the following problem is PTIME-complete: given a graph T and a node x, decide if Alice has a winning strategy in the Win-Move game starting at x.

Solution: By reduction from the circuit problem. The OR-nodes are nodes for Alice to move, and the AND-nodes are those for Bob. TO FINISH.

(e) (10 points) Consider the following startified datalog program that computes the complement of the transitive closure:

```
T(x,y) :- R(x,y)
T(x,y) :- T(x,z),T(z,y)

CT(x,y) :- Node(x),Node(y),not T(x,y)
```

Write a datalog program that computes the complement of the transitive closure assuming inflationary fixpoint semantics. Note: this problem is challenging. You may want to google for the answer, but make sure you understand it.

(f) (10 points) This is a mini literature survey puzzle. Every stratified datalog program can be expressed in inflationary datalog: this is not obvious at all, but the main step in the proof to show that the complement of a datalog program can be computed in inflationary datalog, and this is in essence what you showed in your answer to the previous question. So by now you know that stratified datalog can be translated to inflationary datalog. In this problem we are concerned with the other direction. Many years ago, a researcher (or set of researchers) published a paper where she/he/they proved that the converse also holds: every inflationary datalog program can be rewritten into a stratified datalog program. Let's call this paper and its author(s) X. However, another researcher (or other reserchers) found a flaw in the proof, and published another paper that presented a datalog program Z in inflationary datalog, and proved that Z cannot be expressed in stratified datalog. Let's call this paper and its author(s) Y. Thus, X was (were) wrong! Stratified datalog is strictly less expressive than inflationary datalog, and the query that separates the two classes is Z. In this question you will determine X, Y, and Z. You need to google a lot, but your answer will be very short: you write the titles/authors of the papers X and Y, and the name of the datalog program Z (you don't have to write the program but only say what it does).

Solution:

X is:

Ashok K. Chandra, David Harel: Horn Clauses Queries and Generalizations. J. Log. Program. 2(1): 1-15 (1985)

Y is:

Phokion G. Kolaitis: The Expressive Power of Stratified Programs. Inf. Comput. 90(1): 50-66 (1991)

Z is:

Game tree, or, what we call in this homework the Win-Move game.

2 Conjunctive Queries

2. (60 points)

   (a) (60 points)

   i. For each pair of queries q, q' below indicate whether $q \subseteq q'$. If the answer is yes, provide a proof; if the answer is no, give a database instance I on which $q(I) \not\subseteq q'(I)$.

   1.

$$\begin{aligned} q(x) &:- R(x,y), R(y,z), R(z,x) \\ q'(x) &:- R(x,y), R(y,z), R(z,u), R(u,v), R(v,z) \end{aligned}$$

   2.

$$\begin{aligned} q(x,y) &:- R(x,u,u), R(u,v,w), R(w,w,y) \\ q'(x,y) &:- R(x,u,v), R(v,v,v), R(v,w,y) \end{aligned}$$

   3.

$$\begin{aligned} q() &:- R(u,u,x,y), R(x,y,v,w), v \neq w \\ q'() &:- R(u,u,x,y), x \neq y \end{aligned}$$

   4.

$$\begin{aligned} q(x) &:- R(x,y), R(y,z), R(z,v) \\ q'(x) &:- R(x,y), R(y,z), y \neq z \end{aligned}$$

ii. Consider the two conjunctive queries below, and notice that $q_1 \subset q_2$.

$$q_1(x) = R(x, y), R(y, z)$$
$$q_2(x) = R(x, y)$$

1. Find a conjunctive query $r(x)$ s.t. $q_1 \subset r \subset q_2$

Solution: $r(x) = R(x, y), R(u, v), R(v, w)$

2. **Challenging question** Extend your answer: find an infinite set of queries $r_1(x), r_2(x), \ldots$ such they are inequivalent $r_i \not\equiv r_j$ for $i \neq j$, and for every i, $q_1 \subset r_i \subset q_2$. If you answer this question then you do not need to answer the preceding question.

Solution: $rk(x) = R(x, y_1), R(x_1, y_1), R(x_1, y_2), \ldots, R(x_k, y_k), R(y_k, z)$

3 Transactions

3. (30 points)

(a) (10 points) A *static database* is a database where no insertions or deletions are performed. A *dynamic database* is a database that allows insertions and deletions. We consider a scheduler that has one lock for each record in the database (like SQL Server). Answer the questions below:

i. In a static database, strict two phase locking guarantees that the schedule is serializable while two phase locking does not.

i. **false**

True or false

ii. Strict two phase locking guarantees that the schedule is recoverable, while two phase locking does not.

ii. **true**

True or false

iii. In a dynamic database, strict two phase locking can prevent phantoms, while two phase locking cannot.

iii. **false: needs table lock**

True or false

iv. Strict two phase locking is more difficult to implement and most database system do not support it.

iv. **false**

True or false

v. Strict two phase locking holds all the locks until the end of a transaction, while two phase locking may release the locks earlier.

v. **true**

True or false

vi. In both two phase locking and strict two phase locking all locks must precede all unlocks.

vi. **true**

True or false

vii. In strict two phase locking deadlocks are not possible, while in two phase locking deadlocks are possible.

vii. **false**

True or false

viii. If the database uses shared locks for read operations, then if all transactions are read-only then no deadlocks are possible.

viii. **true**

True or false

ix. SQL Server checks for deadlocks at regular intervals, and if it detects a deadlock then aborts a transaction

ix. **true**

True or false

x. Suppose that the table R has 1000 records of which 50 have $A =' abc'$. Then the transaction below needs to acquire 1000 locks (assuming the database system has locks only for records, i.e. it has not table lock):

```
begin transaction;
select * from R where A='abc';
commit;
```

x. **true**

True or false

(b) (10 points) i. R&G, pp. 598, Exercise 18.4 Consider the execution shown in the following table (same as Fig. 18.7 in the book).

LSN	Action	pLSN	uLSN
00	update: T1 writes P2		
10	update: T1 writes P1		
20	update: T2 writes P5		
30	update: T3 writes P3		
40	T3 commit		
50	update: T2 writes P5		
60	update: T2 writes P3		
70	T2 abort		

Expand the figure to show `prevLSN` and `undonextLSN` values. Then show the actions taken to rollback transaction T2. Finally, show the log after T2 is rolled back, including all `prevLSN` and `undonextLSN` values in the log records.

ii. R&G, Exercise 18.5 Consider the execution showin in the table below (same as Figure 18.8 in the book):

LSN	Action	pLSN	uLSN
00	begin_checkpoint		
10	end_checkpoing		
20	update: T1 writes P1		
30	update: T2 writes P2		
40	update: T3 writes P3		
50	T2: commit		
60	update: T3 writes P2		
70	T2: end		
80	update: T1 writes P5		
90	T3: abort		
	CRASH, RESTART		

In addition, the system crashes during recovery after writing two log records to stabel storage and again after writing another two log records.

1. What is the value of the LSN stored in the master log record?

2. Whas is done during Analysis?

3. What is done during Redo?

4. What is done during Undo?

5. Show the log when recovery is complete, including all non-null prevLSN and undonextLSN values in the log records.

(c) (10 points) For each of the statements below indicate whether it is true of false about the ARIES recovery manager:

i. During the normal operation of a database (i.e. not during recovery) no update records in the log are undone.

i. **false**

true or false ?

ii. During the normal operation of a database (i.e. not during recovery) no update records in the log are redone.

ii. **true**

true or false ?

iii. During the normal operation of a database (i.e. not during recovery) no CLR records are ever written to the log.

iii. **false**

true or false ?

iv. During recovery from a crash no update record is processed both during the redo and during the undo phase. (In other words, an update record is either redone or undone, but not both.)

iv. **true**

true or false ?

v. During recovery from a crash the CLR records in the log are neither undone nor redone. (In other words the CLR records server a different purpose, not to to be redone or undone during recovery.)

v. **false**

true or false ?

vi. Suppose that two update log entries were written in the log on behalf of a transaction, then the system crashed while the transaction was active. Even if the system crashes repeatedly during recovery, there will never be more than two CLR records written on behalf of this transaction.

vi. **true**

true or false ?

vii. All log entries preceeding the last `begin_checkpoint` can be safely deleted from the log in order to reclaim their disc space.

vii. **false**

true or false ?

4 Provenance

4. (30 points)

(a) Consider a relational database schema $R(A), S(B, C)$. All queries mentioned below are assumed to be monotone queries.

i. (10 points) One of the answers a of a relational query Q has the provenance polynomial $x_1y_1^2 + 2x_2y_1y_2$, where x_1, x_2 are annotations of two tuples in R and y_1, y_2 are annotations of two tuples in S.

1. Assume that the input relations are sets. If we evaluate the query under bag semantics, how many copies of a will be in the query's answer ?

i. **3**

Number of copies of a:

2. Assume that the input relations are bags, where each tuple occurs exactly twice. If we evaluate Q under bag semantics, how many copies of a are there in the query's answer ?

i. **24**

Number of copies of a:

3. What is the smallest number of tuples that need to be removed from the database instance in order to remove a from the answer to Q ?

i. **One:** y_1

Number of tuples to be removed:

4. Here we evaluate the query under set semantics. Suppose that the tuple annotated with y_2 was incorrect (it contains some incorrect value), and should not be in the database. Is the answer a correct, or did it become incorrect ?

i. **yes, a is still an answer**

Is a still an answer?:

ii. (10 points) Consider the following instance:

R

A	
a_1	x_1
a_2	x_2

S

A	B	
a_1	b_1	y_1
a_1	b_2	y_2
a_2	b_2	y_3

For each polynomial below, write a Boolean conjunctive query having that provenance polynomial. Your queries should not have constants or the inequality predicates $\neq, <, \leq$.

$$P_1 = x_1y_1 + x_1y_2 + x_2y_3$$
$$P_2 = x_1y_1^2 + x_1y_2^2 + x_1y_2y_3 + x_2y_2y_3 + x_2y_3^2$$
$$P_3 = x_1^2y_1^2 + x_1^2y_2^2 + 2x_1x_2y_2y_3 + x_2^2y_3^2$$

Solution:

$$q = R(x), S(x, y)$$
$$q = R(x), S(x, y), S(z, y)$$
$$q = R(x), S(x, y), R(z), S(z, y)$$

iii. (10 points) Now we consider arbitrary instances $R(A)$, $S(B, C)$, and we assume that the tuples in R are annotated with variables $x_1, x_2, \ldots$ and the tuples in S are annotated with $y_1, y_2, \ldots$ In each case below give an example of a Boolean conjunctive query whose provenance polynomial P has the property stated below. Your conjunctive query may use the predicate $\neq$.

1. The polynomial P factorizes as $P = P_1 \cdot P_2$ where P_1 is a linear polynomial in $x_1, x_2, \ldots$ and P_2 is a linear polynomial in $y_1, y_2, \ldots$

   iii. _____

   Write a query:

   Solution: $q(z) = R(x), S(y, z)$

2. All monomials in P have the form $x_i y_j y_k$ where $j \neq k$.

   iii. _____

   Write a query:

   Solution: Many solutions are possible. Examples:

   $$q(z) = R(x), S(x, y), S(x, z), y \neq z$$
   $$q(z) = R(x), S(x, y), S(u, y), u \neq x$$
   $$q(x) = R(x), S(y, z), S(u, v), y \neq u$$

3. All monomials in P have the form $x_i x_j y_k$ where $i \neq j$.

   iii. _____

   Write a query:

   Solution: Same here, many solutions are possible, e.g.:

   $$q(z) = R(x), R(y), S(x, y), y \neq y'$$

5 Differential Privacy

5. (20 points)

(a) (20 points) Consider a table of patients from a hospital, which contains $n = 10^6$ patient records. (This is a realistic number for the major hospitals in Seattle: the reason it is so high is that the database stores historical data, i.e. each visit of each patient.) Three different statisticians ask the following queries:

- Statistician 1 wants to compute the number n_1 of patients whose diagnostic is *flue*. Assume that the true answer to this query is: $n_1 = 100,000$.
- Statistician 2 wants to compute the number n_2 of patients whose diagnostic is *appendicitis*. Assume that the true answer to this query is: $n_2 = 1,000$.
- Statistician 3 wants to compute the number n_3 of patients whose diagnostic is *the Meckel syndrome*. Assume that the true answer to this query is: $n_3 = 10$.

To answer these queries, the system uses a differentially private algorithm that adds a random noise to the answer. The noise is computed using the Laplace distribution with parameter b (see the paper) chosen such as to guarantee that each answer is ε-differentially private (see the definition in the paper). We consider three cases: $\varepsilon = 0.1$, $\varepsilon = 0.01$, $\varepsilon = 0.001$. For each of the three levels of privacy answer the following questions:

i. What is a patient's privacy guarantee? In other words, for a given patient, what is the upper bound on the probability ratio between the answer of the algorithm on the database with, or without that patient's record? We assume that each patient has a single record in the database. (You need to simply apply the definition of differential privacy, and use a calculator.)

ii. Compute the parameter b for the Laplacian distribution by the differentially private algorithm. (You need apply directly the formula.)

iii. When the differentially private algorithm answers each query, the perturbed answer $\tilde{n}_i$ that it computes happens to be exactly one standard deviation higher than the mean (i.e. n_i). Compute the answer $\tilde{n}_i = n_i + \sigma$ returned by the algorithm, for each of the three statisticians $i = 1, 2, 3$. (Hint: you need to find out the standard deviation of a Laplacian distribution.)

iv. What is the probability that the algorithm returns an answer that is at least one standard deviation larger than the mean, $\tilde{n}_i \geq n_i + \sigma$? This answer will be the same for all three statisticians. (Hint: don't use Tchebishev's inequality; instead compute a simple integral which gives you the exact answer.)

v. Suppose that each statistician wants to know the answer to his/her query within three standard deviations. What is the probability that the differentially private algorithm will return an answer with the desired precision? Note: here "standard deviation" refers to the binomial random variable $B(n, p)$ that characterizes the property inquired, and not to the Laplacian noise. For example, consider the first statistician, who wants to count the number of patients with

"flue". This number is a binomial distribution $B(n, p)$, where $n = 1M$ and $p = 0.1$, because each patient in our population has a $1/10$ chance of having "flue". The statistician wants to compute the value of $n_1 = 100,00$ within 3 standard deviations of the binomial random variable.

You need to answer each question above for each of the three values of ε. If you don't like numerical computations on a calculator, you may choose to turn only the closed formula (in ε, n, n_i) that represents the answer in the *Generic* ε line.

ε	End user privacy	b	$\tilde{n}_i$			$P(\tilde{n}_i \geq n + \sigma)$	Prob. that $\tilde{n}_i$ is within 3 std.dev
			$\tilde{n}_1$	$\tilde{n}_2$	$\tilde{n}_3$		
0.1							
0.01							
0.001							
Generic ε							

Solution:

ε	End user privacy	b	$\tilde{n}_i$			$P(\tilde{n}_i \geq n + \sigma)$	Prob. that $\tilde{n}_i$ is within 3 std.dev
			$\tilde{n}_1$	$\tilde{n}_2$	$\tilde{n}_3$		
0.1							
0.01							
0.001							
Generic ε	$e^{-\varepsilon}$	$b = \frac{1}{\varepsilon}$	$n + b\sqrt{2}$			$\int_{b\sqrt{2}}^{\infty} \frac{1}{2b} e^{\frac{-x}{b}} = \frac{1}{2} e^{-\sqrt{2}}$	