The assignment is to be turned in before Midnight (by 11:59pm) on February 5th, 2016. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded.

1: Concurrency control (2 points)

Consider the schedule shown at Table 1.

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>start</td>
<td>read X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>start</td>
<td>read Y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>write X</td>
<td>start</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>read X</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>write X</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>Commit</td>
</tr>
<tr>
<td>8</td>
<td>read X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>write Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>write X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Commit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>read Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>write Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Commit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Transaction schedule

(a) What is the equivalent serialization order for this schedule? If no order is possible, you may state so and justify your claim.

**Solution:**
Because the serialization graph contains both edges T1 → T2 and T2 → T1, this schedule is not serializable. Thus, it is not equivalent to any serializable schedule.

(b) Assume that in 2PL, each transaction requests proper locking mode immediately before it executing each action, i.e., S lock for read and X lock for write. Let us remove transaction T1, i.e., all operations of T1, from the schedule in Table 1. Is the resulting schedule in 2PL?

**Solution:**
It is not because T2 has to release its lock on X before getting a lock on Y.

(c) Assume that in 2PL, each transaction requests proper locking mode immediately before it executing each action, i.e., S lock for read and X lock for write. Let us remove transaction T2, i.e., all operations of T2, from the schedule in Table 1. Is the resulting schedule in 2PL?

**Solution:**
Yes. T1 may take a shared lock on Y. Then, T3 will take a shared lock on X, upgrades its lock to
exclusive lock and release its lock before commit. T1 can continue by taking a shared lock on X, upgrade both locks to exclusive locks, and finally release both before commit.

(d) Let us remove transaction T1, i.e., all operations of T1, from the schedule in Table 1. What are the maximum degrees of consistency for T2 and T3 in this schedule?

Solution:
T2 cannot keep a long exclusive lock on X (till the EOT). So, T2 can be in degree 0. Given T2 is in degree 0, T3 can set long shared lock and long exclusive locks on X. So, T3 can be in degree 3.

2: Concurrency control (1 point)

Consider a database DB. DB has two relations R1 and R2. The relation R1 contains tuples t1 and t2 and R2 contains tuples t3, t4, and t5. Assume that the database DB, relations, and tuples form a hierarchy of lockable database elements. Tell the sequence of lock requests and the response of the locking scheduler to the following sequence of request. You may assume all requests occur just before they are needed, and all unlocks occur at the end of the transaction, i.e., EOT.

r1(t1); w2(t2); r2(t3); w1(t4)
[w2(t2) represents the update on t2 by transaction T2.]

Solution:
At the first step, T1 puts a IS lock on the DB and on R1, and an S lock on t1. At step 2, T2 puts an IX lock on the DB and on R1, both of which are compatible with the IS locks already there. T2 also puts an X lock on t2. At step 3, T2 puts an IS lock on the DB and on R2 and an S lock on t3, then releases its locks. At step 4, T1 puts an IX lock on the DB and on R2 and an X lock on t4, then releases its locks.

3: Recovery (2 points)

In this problem, you need to simulate the actions taken by ARIES. Consider the following log records and buffer actions:

<table>
<thead>
<tr>
<th>time</th>
<th>LSN</th>
<th>Log</th>
<th>Buffer actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>update: T1 updates P7</td>
<td>P7 brought in to the buffer</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>update: T0 updates P9</td>
<td>P9 brought into the buffer; P9 flushed to disk</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>update: T1 updates P8</td>
<td>P8 brought into the buffer; P8 flushed to disk</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>begin_checkpoint</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>end_checkpoint</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>update: T1 updates P9</td>
<td>P9 brought into the buffer</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>update: T2 updates P6</td>
<td>P6 brought into the buffer</td>
</tr>
<tr>
<td>7</td>
<td>70</td>
<td>update: T1 updates P5</td>
<td>P5 brought into the buffer</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
<td>update: T1 updates P7</td>
<td>P6 flushed to disk</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>CRASH_RESTART</td>
<td></td>
</tr>
</tbody>
</table>

(a) For the actions listed above, show Transaction Table (XT) and Dirty Page Table (DPT) after each action. Assume that DPT holds pageID and recLSN, and XT contains transID and lastLSN.
Solution:

Let XT denote xact table and DPT denote dirty page table.

After time 0:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 00 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
\hline
\end{array}
\]

After time 1:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 00 & active \\
T0 & 10 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
\hline
\end{array}
\]

After time 2:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 20 & active \\
T0 & 10 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
\hline
\end{array}
\]

After time 3: same as above.

After time 4: same as above.

After time 5:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 50 & active \\
T0 & 10 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
P9 & 50 \\
\hline
\end{array}
\]

After time 6:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 50 & active \\
T0 & 10 & active \\
T2 & 60 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
P9 & 50 \\
P6 & 60 \\
\hline
\end{array}
\]

After time 7:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 70 & active \\
T0 & 10 & active \\
T2 & 60 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
P9 & 50 \\
P6 & 60 \\
P5 & 70 \\
\hline
\end{array}
\]

After time 8:

\[
\begin{array}{|c|c|c|}
\hline
\text{transaction} & \text{lastLSN} & \text{status} \\
\hline
T1 & 80 & active \\
T0 & 10 & active \\
T2 & 60 & active \\
\hline
\end{array}
\quad
\begin{array}{|c|c|}
\hline
\text{page} & \text{recLSN} \\
\hline
P7 & 00 \\
P9 & 50 \\
P5 & 70 \\
\hline
\end{array}
\]

Note: P6 is written out with pageLSN = 80.

(b) Simulate Analysis phase to reconstruct XT and DPT after crash. Identify the point where
the Analysis phase starts scanning log records and show XT and DPT after each action.

Solution:

Analysis phase begins by examining the most recent checkpoint and initializing the XP and DPT
<table>
<thead>
<tr>
<th>LSN</th>
<th>Redone?</th>
<th>Why Not?</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>affected page in DPT but recLSN is greater than 10</td>
</tr>
<tr>
<td>20</td>
<td>No</td>
<td>affected page not in DPT</td>
</tr>
<tr>
<td>30</td>
<td>No</td>
<td>checkpoint</td>
</tr>
<tr>
<td>40</td>
<td>No</td>
<td>checkpoint</td>
</tr>
<tr>
<td>50</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>No</td>
<td>pageLSN 80 is greater than 60</td>
</tr>
<tr>
<td>70</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Redo operations

at the time of the checkpoint, i.e., after time 2 in solution to part (a).

After time 5:

<table>
<thead>
<tr>
<th>XT</th>
<th>transaction</th>
<th>lastLSN</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT</td>
<td>T1</td>
<td>50</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T0</td>
<td>10</td>
<td>active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DPT</th>
<th>page</th>
<th>recLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPT</td>
<td>P7</td>
<td>00</td>
</tr>
<tr>
<td>DPT</td>
<td>P9</td>
<td>50</td>
</tr>
</tbody>
</table>

After time 6:

<table>
<thead>
<tr>
<th>XT</th>
<th>transaction</th>
<th>lastLSN</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT</td>
<td>T1</td>
<td>50</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T0</td>
<td>10</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T2</td>
<td>60</td>
<td>active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DPT</th>
<th>page</th>
<th>recLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPT</td>
<td>P7</td>
<td>00</td>
</tr>
<tr>
<td>DPT</td>
<td>P9</td>
<td>50</td>
</tr>
<tr>
<td>DPT</td>
<td>P6</td>
<td>60</td>
</tr>
</tbody>
</table>

After time 7:

<table>
<thead>
<tr>
<th>XT</th>
<th>transaction</th>
<th>lastLSN</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT</td>
<td>T1</td>
<td>70</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T0</td>
<td>10</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T2</td>
<td>60</td>
<td>active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DPT</th>
<th>page</th>
<th>recLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPT</td>
<td>P7</td>
<td>00</td>
</tr>
<tr>
<td>DPT</td>
<td>P9</td>
<td>50</td>
</tr>
<tr>
<td>DPT</td>
<td>P6</td>
<td>60</td>
</tr>
<tr>
<td>DPT</td>
<td>P5</td>
<td>70</td>
</tr>
</tbody>
</table>

After time 8:

<table>
<thead>
<tr>
<th>XT</th>
<th>transaction</th>
<th>lastLSN</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT</td>
<td>T1</td>
<td>80</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T0</td>
<td>10</td>
<td>active</td>
</tr>
<tr>
<td>XT</td>
<td>T2</td>
<td>60</td>
<td>active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DPT</th>
<th>page</th>
<th>recLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPT</td>
<td>P7</td>
<td>00</td>
</tr>
<tr>
<td>DPT</td>
<td>P9</td>
<td>50</td>
</tr>
<tr>
<td>DPT</td>
<td>P6</td>
<td>60</td>
</tr>
<tr>
<td>DPT</td>
<td>P5</td>
<td>70</td>
</tr>
</tbody>
</table>

(c) Simulate Redo phase: first identify where the Redo phase starts scanning the log records. Then, for each action identify whether it needs to be redone or not.

**Solution:**

Redo starts from the smallest recLSN in DPT at the end of Analysis, i.e., LSN=00. Table 2 shows whether each action needs to be redone and the reason.

(d) Simulate Undo phase: identify all actions that need to be undone. In what order will they be undone?
Solution:
As no transaction committed, all actions will be undone in the decreasing order of LSN. That is UNDO 80, 70, 60, 50, 20, 10, and 00.