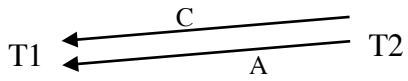
Jonathan Woong
804205763
CS 143
DIS 1B

Homework 6

Problem 1. T1 is the older transaction.

T1	T2
Write D	
	Write C
Read C	
	Write A
Write A	

Q1: is this schedule conflict-serializable? Explain your answer using the precedence graph, and show an equivalent schedule if such a schedule exists.



The schedule is conflict serializable because no cycle exists. The equivalent schedule is T2 → T1

Q2: Could this schedule have been generated by a 2PL protocol? If your answer is no explain why. If your answer is yes, show a 2PL protocol that generated it.

T1	T2
Acquire lock D	
Write D	
	Acquire lock C, Acquire lock A
	Write C
	Unlock C
Acquire lock C	
Read C	
	Write A
	Unlock A
Acquire lock A	
Write A	
Unlock D, Unlock C, Unlock A	

Q3: Say that T2, shortly after the write A operation, checks on some integrity conditions and, at time $\tau 2$, either commits or rollbacks according to the results of those checks. Please answer the following questions:

(i) has T1 read any dirty data?

Jonathan Woong
804205763
CS 143
DIS 1B

Homework 6

Yes. When T1 does Read C, it has read dirty data that may change if T2 decides to rollback.

(ii) what is the earliest time at which T1 can schedule its commit, as to assure recoverability

After write D.

(iii) what actions should actually T1 take depending on whether T2 has committed or aborted?

If T2 commits, T1 will continue normally.
If T2 aborts, T1 must roll back to a state before read C.

Q5: Show what happens under the timestamp-based concurrency control protocol (with Thoma's rule) in the situation where the time stamp of T2 is less than that of T1.

TS(T2) < TS(T1)

T1	T2
TS(T1) < RTS(D)? No TS(T1) < WTS(D)? No Write D	
	TS(T2) < RTS(C)? No TS(T2) < WTS(C)? No Write C
TS(T1) < WTS(C)? No Read C	
	TS(T2) < RTS(A)? No TS(T2) < WTS(A)? No Write A
TS(T1) < RTS(A)? No TS(T1) < WTS(A)? No Write A	

2. The log used by the transaction manager contains information about commits, aborts, updates, and checkpoints. Please answer the following questions about checkpoints as TRUE or FALSE and also give justification for your answer:

1) An important function of checkpoints is to manage the cascading of rollbacks for transactions that had read dirty data from other transactions.

Jonathan Woong
804205763
CS 143
DIS 1B

Homework 6

TRUE: if a transaction is aborted and the log must be recovered, any transaction that reads dirty data from other transactions must be undone in order to ensure that the database reaches a stable and correct state.

2) A key function of checkpoints is to give the database administrator the ability to check on the current state of transactions and decide whether they must abort or commit.

FALSE: The database administrator does not choose which transactions abort or commit. Transactions are scheduled in such a way that they will automatically commit or abort based on integrity checking.

3) The main function of checkpoints is to expedite the recovery process.

TRUE: checkpoints speed up the recovery process by focusing recovery on transactions that occur after the last successful checkpoint (where all changes were committed successfully). In this way, there is no need to recover the entire log in the event of a crash.

4) The transaction manager performs (fuzzy) checkpoint operations by forcing each running transaction to commit and making sure that their updates are actually recorded on disk.

FALSE: the fuzzy checkpoint does not force uncommitted transactions to commit. It only forces the updates of committed transaction into disk.

5) Assume that for unforeseen reasons a few checkpoint records of a lengthy log were damaged in the last crash, and therefore they cannot be read by the recovery manager. Will the recovery manager be able to complete the recovery process anyway and bring back the database to a consistent state, in spite of such losses?

TRUE: the recovery manager can still bring the database to a stable state. It can perform an incomplete recovery by reverting the database to a saved state from a particular date/time.

3. Recovering from a crash using the immediate database modification protocol, the database system finds the following log (stored in three pages).

Checkpoint ---
T0 start

Jonathan Woong
804205763
CS 143
DIS 1B

Homework 6

T0 C, 900, 777
T1 start
Checkpoint ---
----------log page boundary
T2 start
T1 C, 777 , 955
T1 B, 400, 800
T1 commit
T3 start
Checkpoint ---
T2 A, 100, 200
----------log page boundary
T3 B, 800, 600
T4 start
T0 Commit
T4 C, 955, 2100
T5 start
T4 commit
T5 C, 2100, 4000

Q1: How many pages of the log will the system read for recovery (explain)?

2 pages. The system must begin recovery starting from the last-checkpoint (after T3 start and before T2 A, 100, 200), since this checkpoint signals that all commits before this checkpoint have been written to disk.

Q2. Which transactions will be undone and which will be redone?

Undo:
T5 start
T5 C, 2100, 4000
Redo:
T2 A, 100, 200
T3 B, 800, 600
T4 start
T0 Commit
T4 C, 955, 2100
T4 Commit

Q3. What are the values of A, B and C be at the end of a successful recovery (explain the reason for your statement)?

Jonathan Woong
804205763
CS 143
DIS 1B

Homework 6

Use the algorithm starting from the last checkpoint:

Whenever a record *<Ti, Xj, V1, V2>* is found, redo it by writing *V2* to *Xj*

A = 200
B = 600
C = 2100 (because T5 is in the undo list, C will not be 4000)