

CS 431 Fall 2021 Homework 6

1. The operations that will get undone are T3 and T4, the reason why is because a rollback is called on T3 and the changes that were made on T4 were not committed.
2. The operation that needs to be redone is T1 because T3 writes to Y but then gets rolled back, however, T1 then writes to Y, therefore, T1 needs to be redone.
3. The operation that does not need to be redone is T2 because it was committed before the checkpoint.
4. The final value of X is “current” because that was the change that was made in T1 and then committed, however, the change made to X in T4 is not committed.
5. The reason why there aren't log records for read operations because the undo-redo recovery depends on the write operations therefore we need to keep track of the data that was changed.
6. When T1 tries to get a write lock, that means T1 can only transaction that can do Write(B), which means T2 cannot perform Write(B) until T1 is finished.
7. When T2 tries to get a write lock, that means T2 can only transaction that can do Write(B), which means T1 cannot perform Write(B) until T2 is finished.
8. The reason why T1 gets to resume after T2 does its commit is so that it can get the original value of A instead of getting the changed value of A from T1.
9. When T1 tries to get a write lock it means that T1 is the only transactions that can perform Write(A), whereas the other transactions can't perform the write operation on A
10. Using the Would-Wait algorithm to prevent deadlock, assuming that T1 requests a lock which is currently being held by T2 then it means that T1 can wait because it is a younger process until T2 finishes. This allows T1 to rollback if its timestamp happens to be greater than T2. I drew an illustration to demonstrate the Would-Wait algorithm between T1 and T2.

