Q1 Construct an example of some transactions and a non-serializable interleaved execution H of those transactions, such that the precedes graph of H has no cycle of length 2.

Q2 Either prove or disprove the following claim: if H is an execution such that no pair of concurrently active transactions have a ww-conflict on the same item, then H is serializable.

Q3 Consider a DBMS which uses multigranularity locking (section 16.4 of Silberschatz-Korth-Sudarshan 5th ed) with rigorous two-phase locking. If H is an execution of the system, and the precedes graph of H has an edge from Ti to Tj, prove that commit(Ti) occurs before commit(Tj) in H.

Q4 Fred Foolish is building a DBMS which uses multi-granularity locking (the granules are files and records). Fred makes a mistake in his coding, so that S locks and IX locks on a granule are not conflicting in his system (all other pairs of locks are correctly checked as given in Fig 16.17 of Silberschatz-Korth-Sudarshan 5th ed). Is Fred's system satisfactory? If so, show this; if not, give an example of an execution that Fred's system allows, which is not serialisable.

Q5 Show that if each transaction running alone preserves every integrity condition on the database, then each integrity condition is valid after an arbitrary concurrent execution in which each read-only transaction runs with cursor stability, and each update transaction (doing reads and writes) runs with two-phase locking.

Q6 Construct an example of an integrity condition, two transactions T1 and T2, and a concurrent execution H, such that each transaction running alone preserves the integrity condition, T1 is a read-only transaction running with cursor stability, T2 runs with two-phase locking, and the data read by T1 during H does not satisfy the integrity constraint. (Of course, H must be non-serializable, since in any serial execution, T1 sees data that satisfies all integrity constraints.)

Q7 Construct the multiversion conflict graph for the execution H3 from the paper "A Read-Only Transaction Anomaly Under Snapshot Isolation" by Fekete, O'Neil and O'Neil, ACM SIGMOD Record 33(3):12-14, 2004. Also, let H4 denote the subset of H3 consisting of the operations of T1 and T2; draw the conflict graph for H4, and show that H4 is serializable.

Q8 Suppose T1, T2, T3 and T4 are transactions that are each running using SI. Consider an interleaved execution in which the following operations occur in the order shown. Fill in the version information for each accessed

item (for example, the first read operation is $r_1[x_0]$, because the read of x by T1 accesses the initial version of x.)
start T1
T1 reads x
T1 writes x
start T3
commit T1
start T2
start T4
T2 reads x
T2 reads y
T4 reads x
T3 reads x
T2 writes y
T4 reads y
commit T2
T3 reads y
T4 writes x
commit T4
commit T3