Data Management – A.Y. 2013/14 – exam of 15/9/2014

Problem 1
Consider the following schedule

$$S = r_1(X)\, r_3(T)\, r_3(Y)\, w_2(X)\, w_1(Y)\, r_3(Z)\, w_2(Z)\, r_3(X)\, r_2(T)$$

- Tell whether S is conflict-serializable. If the answer is yes, then show a serial schedule that is conflict-equivalent to S. If the answer is no, then explain the answer.
- Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then explain the answer.
- Tell whether S is ACR (Avoid Cascading Rollback) or not, explaining the answer.
- Consider the schedule S' obtained from S by deleting the action $r_3(X)$, and tell whether S' is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then explain the answer.

Problem 2
Suppose that S is a complete schedule that

- is accepted by the timestamp-based scheduler, and
- does not contain more than one write operations on the same database element.

Prove or disprove that S is view serializable.

Problem 3
Consider the relation `TICKET(pcode,theater,date,cost)`, where each record stores information about a ticket sold to a person for a theater show, with the code of the person who bought the ticket, the theater, the date, and the cost of the ticket. The relation has 3.000.000 tuples, stored in 150.000 pages, and has 10.000 different values in the attribute `theater`. We assume that all fields in every record have the same length, including the pointers. There is a dense, non-clustering B^+-tree index on `TICKET` with search key `theater`, using alternative 2. Consider the query that asks for date and cost of all tickets sold for a given theater, and tell how many page accesses are needed for computing the answer to the query.

Problem 4
Suppose we have the relations `R(A,B)` and `S(B,C)`, where `R` is constituted by 5000 pages, and `S` is constituted by 2000 pages, and we have 102 buffer pages available. What is the cost (in terms of number of pages that are either read or written) of the natural join between `R` and `S` if we use the page-based nested loop join algorithm with `R` as the outer relation? Explain in detail your answer. And what is the cost of the block nested loop join algorithm with `R` as the outer relation. Explain in detail your answer.

Problem 5
The relation `Actor` is stored in 70.000 pages, the relation `Director` is stored in 20.000 pages, and there is an effective hash function working on the records of the two relations. Describe in detail the algorithm you would use to compute the intersection of the two relations, assuming that 200 buffer pages are currently available. Tell which is the cost of the chosen algorithm in terms of page accesses.