Data Management – AA 2013/14 – exam of 4/7/2014

Problem 1
Let S, S' be two complete schedules on transactions T_1, T_2, T_3, T_4, T_5, let $r_5(x)$ precede $w_3(x)$ in S, and let $w_4(x)$ be either before $r_5(x)$, or after $w_3(x)$ in S. Prove or disprove each of the following two statements:

1. if S' is view-equivalent to S, then $w_4(x)$ is either before $r_5(x)$, or after $w_3(x)$ in S' too.
2. if S' is view-equivalent to S, then S' is not accepted by the 2PL schedule with shared and exclusive locks.

Problem 2
Let S be the following schedule:

$$w_3(z)\, r_1(y)\, w_2(x)\, w_3(y)\, r_3(x)\, r_2(z)\, w_1(z)\, w_3(t)$$

- 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 or not.
- Tell whether S is recoverable or not.

Problem 3
Consider the following schedule S:

$$w_1(z)\, r_2(y)\, w_3(x)\, w_1(y)\, r_1(x)\, r_3(z)\, w_2(z)\, w_1(t)$$

and illustrate the various steps carried out by the timestamp-based scheduler when analyzing the above schedule.

Problem 4
The relation `Player` is stored in 60.000 pages, and the relation `Trainer` is stored in 30.000 pages. Describe in detail the algorithm you would use to compute the union of the two relations, assuming that 100 buffer pages are currently available. Tell which is the cost of the chosen algorithm in terms of page accesses.

Problem 5
Consider the relation `Professor(lastname, firstname, dateofbirth, cityofbirth, university, salary)`, which store information about professors. The most frequent queries posed to such relations are:

- list last name and first name of all professors, without duplicates;
- given last name, first name, and date of birth, list the city of birth, and the salary of all professors with the given last name, first name and date of birth.

We know that relation `Professor` has 15.000.000 tuples, the size of every memory page is 18.000 B, the size of each field (relation attribute, or pointer) is 30 B, and there are in the average 5 professors with the same combination of last name, first name, and date of birth.

Tell which method would you choose for representing the relations in secondary storage, taking into account that your goal is to execute the above queries efficiently. Also, for each query, tell which algorithm would you choose for executing the query, and how many page accesses would be needed for computing the answer to the query, given the chosen algorithm, and assuming that 1520 buffer pages are available.