Data Management – exam of 03/06/2013

Problem 1 Consider the following schedule

$$S = r_1(A)\, r_2(A)\, w_1(A)\, r_2(B)\, w_3(D)\, r_3(C)\, r_1(C)\, r_2(C)\, r_4(A)\, w_3(C).$$

1. 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.
2. 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.

Problem 2 Prove or disprove the following facts:

1. Every strict schedule is conflict-serializable.
2. Every schedule following the strong strict 2PL protocol is rigorous.
3. Every schedule following the strong strict 2PL protocol is serializable.

Problem 3 Explain the difference between the two buffer policies "steal" and "no-steal", as well as the difference between the two buffer policies "force" and "no-force".

Problem 4 Suppose that page P in our Data Base Management System is a page with fixed-length records containing 100 slots, and suppose we ask for the insertion of a new record. Illustrate the various actions that the system performs in the two cases of packed and unpacked organization for P, respectively.

Problem 5 Consider the relation `Player(code,age,team)`, where `code` is the primary key of the relation, and for which the relevant queries are the following:

1. Query **Q1**: given a value t for team, and a range r of age, return all players whose team is t, and whose age falls into range r;
2. Query **Q2**: given a code d, return the team of the player whose code is d.

We know that the size of each attribute value is 4 bytes, the size of each page-id is also 4 bytes, the size of each page is 1.200 bytes, and the number of records of the relation `Player` is 66.666.600.

1. Tell which is the method you would choose for representing the relation `Player`, explaining your decision.
2. On the basis of the method chosen for representing the relation `Player`, tell which is the cost of the execution of query **Q1**, expressed in terms of page accesses.