Data Management – AA 2013/14 – exam of 31/1/2014

Compito A

Problem 1
Consider the case of secondary storage pages with fixed length records, and describe in detail the difference between the packed and unpacked organizations.

Problem 2
A relation R has 9.000 pages, and our DBMS has 101 buffer frames free. Describe in detail the algorithm you would use to sort R under this condition, and tell which is the cost of the algorithm in terms of page accesses.

Problem 3
Consider the relation `RENT(code,person,city,cost)` that stores information about rents of bicycles, with the code of the bicycle, the person who rented the bicycle, the city of the rent, and the cost of the rent. The relation has 2.000.000 tuples, stored in 200.000 pages, and has 10.000 different values in the attribute `cost`. We assume that all fields in every record have the same length, independently of the attribute. There is a dense, non-clustering B^+-tree index on `RENT` which search key `cost`, using alternative 2. Consider the query that asks for code, person, and city of all rents with a given cost, and tell how many page accesses we need for computing the answer to the query.

Problem 4
Consider the following schedule

$$S = r_1(x)\, r_2(y)\, w_1(z)\, w_2(y)\, w_3(x)\, r_1(y)\, r_3(z)\, w_4(y).$$

4.1 Tell whether S is serializable or not, explaining the answer in detail.
4.2 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.
4.3 Tell whether S is a strict 2PL schedule, explaining the answer in detail.
4.4 Tell whether S is a strong strict 2PL schedule, explaining the answer in detail.

Problem 5
A "schedule with tight locks" is a schedule with read, write, lock (both shared and exclusive) and unlock commands, that satisfy the following properties:

- it is legal,
- all of its transactions are well-formed,
- every $sl_i(x)$ command appearing in it is immediately followed by the $r_i(x)$ command, and
- every $xl_i(x)$ command appearing in it is immediately followed by the $w_i(x)$ command.

Prove or disprove that every schedule with tight locks whose associated wait-for graph is cyclic is not conflict-serializable.