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

Compito B

Problem 1
Illustrate the definition of "heap", and describe in detail the method for representing a heap through linked lists, and the method for representing a heap through directory.

Problem 2
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.

Problem 3
Consider the following schedule

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

3.1 Tell whether S is serializable or not, explaining the answer in detail.
3.2 Tell whether S is a 2PL schedule, explaining the answer in detail.
3.3 Tell whether S is a strict 2PL schedule, explaining the answer in detail.
3.4 Tell whether S is a strong strict 2PL schedule, explaining the answer in detail.

Problem 4
Consider the relation `FLIGHT(code,pilot,airplane,length)` that stores information about flights, with the code of the flight, the name of the pilot, the airplane used by the flight, and the length of the flight in Kilometers. The relation has 1.500.000 tuples, stored in 150.000 pages, and has 10.000 different values in the attribute `length`. We assume that all fields have the same length, independently of the attribute. There is a dense, non-clustering B^+-tree index on `FLIGHT` which search key `length`, using alternative 2. Consider the query that asks for code, pilot, and airplane of all flights with a given length, and tell how many page accesses we need for computing the answer to the query.

Problem 5
A relation R has 12.000 pages, and our DBMS has 11 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.