Data Management – AA 2015/16 – exam of 13/06/2016

Problem 1
Describe one possible method that the DBMSs can use to solve the problem of storing a tuple of a relation that does not fit in one page of secondary storage.

Problem 2
A relation R with attributes A, B, C is stored in 6.000 pages, a relation S with attributes D, E (where D is a key of S) is stored in 3.000 pages, and our DBMS has 102 buffer frames free. We want to compute the join of R and S on the condition $C = D$.

1. Can we use the block nested-loop algorithm?
2. Can we use a two-pass algorithm?

For each of the two cases (1 and 2), (i) if the answer is no, then explain the answer; (ii) if the answer is yes, then describe in detail how the algorithm works, and tell which is the cost of the algorithm for computing the above join in terms of number of page accesses.

Problem 3
Consider the relation `TRAVEL(code,person,nation,cost)` that stores information about travels of people, with the code of the travel, the person who traveled, the nation visited, and the cost of the travel. The relation has 3.000.000 tuples, stored in 300.000 pages, and has 10.000 different values in the attribute `cost`, uniformly distributed in the relation. We assume that all fields and pointers have the same length, independently of the attribute. There is a sparse, clustering B^+-tree index on `TRAVEL` with search key `cost`, using alternative 2, with one data entry for each page of `TRAVEL`. Consider the query that asks for code and nation of the travels whose cost is in a given range constituted by 10 values, and tell how many page accesses we need for computing the answer to the query.

Problem 4
Consider the following schedule

$$S = r_1(z)\, r_2(z)\, r_3(x)\, w_3(z)\, w_4(x)\, r_1(x)\, w_2(x)\, w_2(v)\, w_5(v)\, w_5(y)\, r_1(y).$$

4.1 Tell whether S is conflict-serializable or not, explaining the answer in detail.
4.2 If S is conflict-serializable, then describe all the serial schedules that are conflict-equivalent to S, otherwise tell which is the minimal number of transactions that must be deleted from S so that the resulting schedule is conflict-serializable.
4.3 Tell whether S is view-serializable or not, explaining the answer in detail.
4.4 Tell whether S is in ACR (Avoid Cascading Rollback) or not, explaining the answer in detail.

Problem 5
Let S be a schedule that is constituted only by write actions, and such that every command of type $xl_i(x)$ in S appears immediately before the corresponding $w_i(x)$ action. Prove or disprove each of the following two claims:

- If S contains exactly *two* transactions, and is not accepted by the 2PL scheduler because of the presence of a deadlock, then it is not view-serializable.
- If S contains exactly *three* transactions, and is not accepted by the 2PL scheduler because of the presence of a deadlock, then it is not view-serializable.