Data Management – AA 2015/16 – exam of 08/1/2016

Problem 1
Let `R` be a relation all of whose attributes have a fixed length, and such that no deletion is allowed on `R`. Which page organization method would you choose for storing the records within each of the pages of `R`? Please, explain the answer in detail, pointing out the advantages of the chosen page organization method with respect to other possibile methods.

Problem 2
In general, a secondary, non-unique index contains duplicates (we remind the students that a duplicate is a pair of data entries with the same value for the search key). Are there cases where a secondary, non-unique index does not contain duplicates? If yes, which are those cases? Explain the answer in detail.

Problem 3
Prove or disprove each of the following claims.

1. If S is a schedule in ACR (Avoiding Casading Rollback), then S is view-serializable.
2. If S is a schedule containing at most one write action, then S is conflict-serializable.

Problem 4
Consider the following schedule

$$S = r_2(x)\, r_1(x)\, w_3(t)\, w_1(x)\, r_3(y)\, r_4(t)\, r_2(y)\, w_2(z)\, w_5(y)\, w_4(z)$$

and answer the following questions:

- Tell whether S is conflict serializable, and, if so, exhibit one serial schedule which is conflict-equivalent to S.
- Tell whether S is in 2PL with shared and exclusive locks, motivating the answer.
- Tell whether S is strict, motivating the answer.

Problem 5
Suppose we have to compute the set intersection between the relation `R` with 200.000 pages and the relation `S` with 150.000 pages, having 600 free frames in the buffer. Tell which algorithm would you choose among the following three alternatives: 1 pass, 2 pass, and 3 pass. Also, tell which is the cost of the algorithm you have chosen for computing the result. Explain in detail both answers.

Problem 6
Consider the relations `Tournament(name,year,city,winner)`, which contains 600 pages storing information about 30.000 tennis tournaments, and `Player(code,name,yearOfBirth,cityOfBirth)`, which stores information about 200.000 tennis players. Assume that every attribute and every pointer has the same size, and that there is a tree-based, primary, unclustered index on `Player` with search key `code`. Consider the query

```
select *
from Tournament, Player
where winner = code
```

and, assuming that you can only use one buffer frame (besides what is needed for handling the output), provide the answer to the following two questions, explaining your answers in detail:

1. which algorithm would you use for computing the answer to the above query?
2. which is the cost of computing the answer to the above query using the chosen algorithm?