ACS Theory Assignment 3

This assignment is due via Absalon on Jan 11, 23:59. While individual hand-ins will be accepted in exceptional cases, we strongly recommend that this assignment be solved in groups of maximum two students. A well-formed solution to this assignment should include a PDF file with answers to all theory questions.

All homework assignments have to be submitted via Absalon in electronic format. It is your responsibility to make sure in time that the upload of your files succeeds. While it is allowed to submit scanned PDF files of your homework assignments, we strongly suggest composing the assignment using a text editor or LaTeX and creating a PDF file for submission. Email or paper submissions will not be accepted.

Learning Goals

This assignment targets the following learning goals:

- Carry out data analysis tasks using the algorithms presented in the lecture and analyze their I/O cost.
- Modify data processing algorithms presented in your lecture to fit a given data analysis task.
- Perform simple system reliability calculations, while clearly stating underlying assumptions.

Question 1: Data Processing I

You are asked to analyze the sales records of an e-commerce company. The following tables are provided to you:

- `Orders(oID, pID, quantity)`, which stores the ID of the order (`oID`), the ID and the quantity (`quantity`) of the product (`pID`) in the order.
- `Products(pID, category)`, which stores the ID of each product (`pID`) and its category (`category`).

Your task is to produce sales statistics:

For each category of products, return the total quantities that are sold.

The schema of the results should be `(category, quantity)`.

The first step is **joining** the tables `Orders` and `Products` on `pID`. Suppose you have a server with a buffer size of 500 pages. You consider using Sort Merge Join (SMJ) and Grace Hash Join (GHJ). You want to write an automated algorithm to choose the most efficient algorithm between SMJ and GHJ at runtime according to the size of the two tables. The size of `Products` is fixed and known to be 40 000 pages. While the exact size of `Orders` is unknown, it is known to be larger than `Products` in this scenario.

1. What is the range of the size of `Orders` in which SMJ is equally efficient as GHJ in terms of I/O cost? Briefly explain your answer.

   Hint: Consider the number of buffer pages needed to complete each of GHJ and SMJ in two passes.

2. Considering the same sizes as given above, are there sizes of `Orders` for which Block Nested-Loop Join (BNLJ) has lower I/O cost than SMJ? Briefly illustrate your answer.

The next step is **grouping** the records resulting from the previous step by their `category` values and **calculating the sum** of `quantity` values over each group. The final results should be sorted by `category`. *Suppose the previous step produces join results of* 200 000 *pages, and you have* 1 000 *machines each with a buffer of* 500 *pages.* Assume the join results have already been *randomly* partitioned and stored in the 1 000 machines.

3. Describe the most efficient parallel algorithm (in terms of network I/O) that can make use of all the 1 000 machines to carry out this step. What is the cost in terms of network I/O? Briefly explain your answer.

Question 2: Data Processing II

An online gaming company, E-Game, maintains an operational database storing information on their players. The management of E-Game would like to analyze players. They assign the following data analysis task to you:

> Group players by country and list all players with: `player_id`, country, ranking within their country, and also report the average points of all the players in that country.

The ranking of a player is based on their points, i.e., the one with the highest points is ranked 1st, the second highest is ranked 2nd, and so on. Since the analysis is heavy, to avoid interference with other operations, the data necessary for your task has been pre-extracted outside of the database and stored separately. You are provided with a heap file storing a table with schema: `Players (player_id, points, country)`.

You are provided with one server with memory larger than the *square root* of the size of the player data by at least one page, but insufficient to hold them completely. Therefore, you have to resort to external memory algorithms. However, as the number of countries is relatively small, you may assume that you can store statistics on all countries in memory

without affecting the efficiency of the external memory algorithms. If you need to make any further assumptions, please state them clearly as part of your solution. Please answer the following questions.

1. State an external memory algorithm with the minimum I/O cost to answer the above query. Argue for the algorithm's correctness and efficiency.

2. State the I/O cost of the algorithm you designed. You can make your own notation for the size of each attribute in any table.

Note 1: To state an algorithm, you can reference existing sort-based or hash-based external memory algorithms. You should not state all the steps of existing algorithms from scratch, but you should clearly state the steps that you need to change in the algorithms you reference, and how you change these steps. To describe how you change a step, refer to the step and list the sub-steps that need to be executed to achieve your goal.

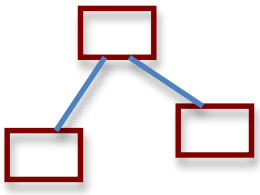
Note 2: Instead of using several existing external memory algorithms in sequence, you should design a single algorithm that addresses the whole task holistically. That is why **Note 1** asks you to show changes to steps of existing algorithms, if necessary.

Note 3: Your solution will be graded partly for the efficiency of your algorithm and partly for the quality of the justifications. That is to say, a solution with justified lower I/O cost will help you get a better grade.

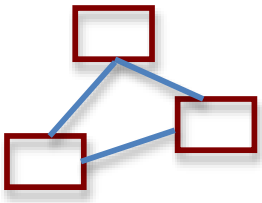
Question 3: Reliability

The town council wants to implement a municipal network to connect the local area networks in the library, the town hall, and the school. They want to minimize the chance that any building is completely disconnected from the others. They are considering two network topologies:

1. Daisy Chain:



2. Fully connected:



Each link in the network has a failure probability of p. Now, answer the following questions:

1. What is the probability that the daisy chain network is connecting all the buildings?

2. What is the probability that the fully connected network is connecting all the buildings?

3. The town council has a limited budget, with which it can buy either a daisy chain network with two high reliability links ($p = .000001$), or a fully connected network with three low-reliability links ($p = .0001$). Which should they purchase?

Remember to explicitly state the assumptions you make for your reliability calculations.