Mandatory Exercise 2

IN3020&IN4020 (2026)

Deadline: 18 March 2026, 23:59 (Devilry)

Question 1.

Consider two relations with the schemas MathStudents[SID, SName, Age] and CSStudents[SID, SName, Age]. Compile the following query to a logical query plan:

```
SELECT * FROM MathStudents
WHERE SName = 'Peter' AND NOT Age > ANY (SELECT Age FROM CSStudents)
```

Question 2.

Consider the following relations, where each SID and each Age is an integer taking 10B, and all other attributes are strings taking 30B:

MathStudents:

SID	SName	Age
1	Peter	40
2	Peter	20
3	Anna	20

CSStudents:

SID	SName	Age
1	Julia	30
2	Peter	20
2	Peter	40

Takes:

SID	Course
1	Databases
2	Astronomy
2	Databases

Consider also the following logical query plan

$$\pi^{\mathsf{b}}_{\mathsf{SName,Course}}((\sigma_{\mathsf{SName='Julia'}}(\mathsf{MathStudents}) \cap \mathsf{CSStudents}) \bowtie \mathsf{Takes})$$

Part a.

Compute the statistics of the relations relevant to our cost function (i.e., $Size(\mathsf{R})$, $Tups(\mathsf{R})$, $TupSize(\mathsf{R})$, and $Vals(\mathsf{R.A})$ for all relations R and their attributes A).

Part b.

Using the cost function introduced in the lectures, compute the cost of this query plan based on the statistics computed in Part a.

Question 3.

Consider a relation, where each tuple takes 300B, and an index for this relation, which has 20B per search key and 4B per pointer. Let both be stored in a continuous unspanned manner in blocks of size 8000B.

Let us call an index more efficient than another (for the same relation) if it uses less block accesses in the worst case; for example, an index requiring at most 4 block accesses is better than an index requiring at most 6 block accesses.

Part a.

What is the minimal number of tuples in the relation such that the static two-level index with dense first level and sparse second level is more efficient than the usual (i.e., one-level) dense index?

Part b.

What is the minimal number of tuples in the relation such that the static two-level index with both levels sparse is more efficient than the usual (i.e., one-level) sparse index?