Chair of applied computer science III

Prof. Dr. Guido Moerkotte

Email: moerkotte@uni-mannheim.de

Query Optimization — Exercise sheet 10

Exercise 1

Exercise 1 a)

Describe the Cheung formula.
Compute the Cheung formula for the values $N = 1000, m = 100, k = 15$

Solution

Read the paragraph on *Selecting k non-distinct items* in the chapter *Counting the Number of Direct Accesses* in the script .
To compute the result, plug in the values in the formula.

Exercise 2

Exercise 2 a)

Read the chapter on *Genetic Algorithms* in the *Probabilistic Algorithms* section in the script.

Exercise 2 b)

Implement the genetic algorithm for the join ordering problem. It is sufficient to consider only left deep trees.
Note, on a high level, it is easy to implement the algorithm. However you will need a significant number of helper functions.
Feel free to take the solution code and only implement the high level level algorithm (`Run` function).

Exercise 3

Queries with multiple predicates may require the evaluation of multiple indices. In what order should you process the predicates?

Solution

- Order predicates by selectivity. Problem: Ignores predicate evaluation cost.

- Better: Order by $(1 - s)/c$, where s is selectivity and c is evaluation cost of a predicate.

- The above is optimal if the predicates are independent.

Exercise 4

Instead of using B-Tree one can also use hashing based data structures. For instance, one can build on the idea of extendible hashing
`https://en.wikipedia.org/wiki/Extendible_hashing`
For queries with what type of predicates can you apply such data structures?

Solution

- Only point queries (e.g.: R.a = 5)

- In principle, can be used for hash joins on base relations.

Exercise 5

You are given a relation with 6 tuples. These tuples are equally distributed over 3 pages. Compute the average number of page accesses for reading 2 tuples. Assume that all tuples have the same probability to be read.

Solution

Yao
Yao assumptions: m buckets with n items each. Randomly select k distinct items.
Yao gives us the average number of qualifying buckets.
Buckets correspond to pages and items to tuples.
Plug in the above values and obtain $\overline{\mathcal{Y}} = 1.8$.

Detailed Explanation, no Yao
There are 2 tuples per page. Think of an urn model with 2 blue, 2 red and 2 green balls. Each colour refers to a page. We want to draw 2 balls at once. This is equivalent to drawing two balls sequentially without replacement. We draw the first ball and it has *some* colour. Then, there are 5 balls left. The probability that the second ball has the same colour is 1/5. Hence, the probability that the second ball has a different colour is 4/5.

Only when we draw the same colour twice we have one page access. Otherwise we have two page page accesses. It follows that the average number of page accesses is

$$\frac{1}{5}1 + \frac{4}{5}2 = 1.8.$$

Exercise 6

BONUS

Write a program that allows you to measure some physicial properties of your disk.

Solution

Code: SimpleBenchmark.java

- OS: openSuSe 12.2
- Sequential write: 10.6 MB/s (1.5 ms/page)
- Sequential read: 31.0 MB/s (0.5 ms/page)
- Random write: 2.1 MB/s (7.4 ms/page)
- Random read: 2.3 MB/s (6.7 ms/page)