December 8, 2020
Arnab Chakrabarti, Rihan Hai, Stefan Braun

Implementation of Databases (WS 20/21)

Exercise 3

Due until December 17, 2020, 10am.
Please submit your solution *in a single PDF file* before the deadline to the RWTHmoodle system!
Please submit solutions in groups.

Exercise 3.1 (Sorting) (5 pts)

Suppose you have a file of 15,000 pages and eight buffer pages and you are sorting it using general (external) merge-sort. Please answer the following questions:

1. How many runs will you produce? Remark: When a file is sorted, in intermediate steps subfiles are created. Each sorted subfile is called a *run*.
2. How many passes will it take to sort the file completely?
3. How many buffer pages do you need at least to sort the file in two passes?
4. How many runs and passes would a Two-Way-Sort algorithm take?

Exercise 3.2 (Join) (5 pts)

Given are the two relations Album and Tracks, with the following specifics:

- Album has a size of 10.000 pages, 40 bytes record size and 100 tuples/page
- Track has a size of 200.000 pages, 30 bytes record size and 80 tuples/page

You have 16 buffer pages available.

1. Calculate the I/O requirements of a simple nested loop join
2. Calculate the I/O requirements of a block nested loop join
3. Explain the differences between the two algorithms. What are the similarities and differences? How does the block nested loop join reduce I/O costs?

Exercise 3.3 (Quant Graphs) (8 pts)

Given is the following relational database schema:

- $Prof(\underline{ssn}, pname, city)$: Professors with social security number (SSN), name, and city
- $Course(\underline{ctitle}, ssn, dnr)$: Courses with titles, the SSN of the lecturer, and number of the department

- $Dept(\underline{dnr}, dname, city)$: Departments with number, name, and city
- Interrelational dependencies: (foreign keys)
  - $Course[ssn] \subseteq Prof[ssn]$
  - $Course[dnr] \subseteq Dept[dnr]$

1. Specify the following query in the tuple relational calculus (TRC) and draw the corresponding quant graph:

   Names of professors that give lectures in two different departments.

2. The following query is given as a tableau. Translate it into the tuple relational calculus (TRC) and draw the quant graph for this query.

TAG	ssn	name	city	ctitle	dnr	dname
$TARGET$		n_1				
$Prof$	s_1	n_1	c_1			
$Course$	s_1			t_1	d_1	
$Dept$			c_1		d_1	n_2

3. Check whether the quant graphs contain a (predicate) cycle.

Exercise 3.4 (Tableau Containment and Minimization) **(9 pts)**

Given are the following tableaus:

T_1

a1	a2	
b3	a2	(R)
a1	b4	(R)
5	b3	(R)
b4	5	(R)

T_2

a1	a2	
b4	a2	(R)
b1	a2	(R)
a1	b3	(R)
b2	b4	(R)
b2	b1	(R)
b3	b2	(R)

T_3

a1	a2	
b3	a2	(R)
a1	b2	(R)
b4	b1	(R)
b1	b2	(R)
b2	b3	(R)
b1	b3	(R)

1. Find out if $T_i \subseteq T_j$ i.e., $T_i \equiv T_j$ for $i \neq j, i, j \in \{1, 2, 3\}$.

2. Write down the minimal tableau for $T_i, i \in \{1, 2, 3\}$.

Exercise 3.5 (Cost Estimation) **(13 pts)**

Consider a relation $R(a, b, c, d)$ containing 5,000,000 records, where each data page of the relation holds 10 records.

1. Suppose R is organized as a sorted file with indexes, and R is stored in $R.a$ order. There are three access paths:

   A1. Access the sorted file for R directly.

   A2. Use a clustered B+ tree index on attribute $R.a$.

   A3. Use a clustered hash index on attribute $R.a$.

   For each of the following selection queries, state which of the three access paths is most likely to be the cheapest and explain why.

   (a) $\sigma_{a=50{,}000}(R)$

   (b) $\sigma_{a\neq 50{,}000}(R)$

   (c) $\sigma_{a>50{,}000 \wedge a<50{,}010}(R)$

2. Assume that all four attributes of R are string fields of the same length. There are 1000 buffer pages. A projection query $\pi_{a,b}(R)$ should be executed. 20 records of the resulting relation can be stored in one page. Consider the optimized version of the sorting-based projection algorithm: The initial sorting pass reads the input relation and creates sorted runs of tuples containing only attributes a and b. Subsequent merging passes eliminate duplicates while merging the initial runs to obtain a single sorted result (as opposed to doing a separate pass to eliminate duplicates from a sorted result containing duplicates).

   (a) How many sorted runs are produced in the first pass? What is the average length of these runs? (Assume that memory is utilized well and any available optimization to increase run size is used.) What is the I/O cost of this pass (**including** writing of the intermediate result)?

   (b) How many additional merge passes are required to compute the final result of the projection query? What is the I/O cost of these additional passes? Writing of final result is **excluded**.

3. Suppose the following query with a self-join on R is executed: $\pi_{a,b}(R \bowtie_{a=a} R)$. Note that a is not a key attribute. Which join method (block-nested loop join, hash join, or sort-merge join) using any of the access paths from above is most likely to perform best. Why? What are the estimated costs?