Database Management Systems

Practice Problem Set: Query Evaluation, Optimization

---

1. Suppose that a relation R(*a,b,c,d,e*) containing 5,000,000 records, where each data block of the relation holds 10 records, is organized as a sorted file with secondary indexes. Assume that *R.a* is a candidate key for R, with values lying in the range 0 to 4,999,999, and that R is stored in *R.a* order. Consider the following alternative strategies for evaluating a selection query over R:

- Access the sorted file for R directly.
- Use a clustered B+ tree index on attribute *R.a*.
- Use a hashed index on attribute *R.a*.
- Use a clustered B+ tree index on attributes (*R.a, R.b*).
- Use a hashed index on attributes (*R.a, R.b*).
- Use an unclustered B+ tree index on attribute *R.b*.

For each of the following relational algebra queries, state which of the above evaluation strategies is likely to be the cheapest:

1. $\sigma_{a<50{,}000 \wedge b<50{,}000}(R)$
2. $\sigma_{a=50{,}000 \wedge b<50{,}000}(R)$
3. $\sigma_{a>50{,}000 \wedge b=50{,}000}(R)$
4. $\sigma_{a=50{,}000 \wedge a=50{,}010}(R)$
5. $\sigma_{a\neq50{,}000 \wedge b=50{,}000}(R)$
6. $\sigma_{a<50{,}000 \vee b=50{,}000}(R)$

2. Consider the join $R \bowtie_{R.a=S.b} S$ given the following information about the relations to be joined. The cost metric is the number of block I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

Relation R contains 10,000 tuples and has 10 tuples per page.
Relation S contains 2000 tuples and also has 10 tuples per page.
Attribute *b* of relation S is the primary key for S.
Both relations are stored as simple heap files.
Neither relation has any indexes built on it.
52 buffer pages are available.

i. What is the cost of joining R and S using a nested loop join?
ii. What is the cost of joining R and S using a block nested loops join?
iii. What is the cost of joining R and S using a hash join?

3. Consider processing the following SQL projection query:

SELECT DISTINCT E.title, E.ename FROM Executives E

You are given the following information:

Executives has attributes *ename*, *title*, *dname*, and *address*; all are string fields of the same length.
The *ename* attribute is a candidate key.
The relation contains 10,000 pages.
There are 10 buffer pages.

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 *ename* and *title*. 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). The cost metric is the number of page I/Os.

i. How many sorted runs are produced in the first pass?
ii. What is the average length of these runs?
iii. What is the I/O cost of this sorting pass?
iv. How many additional merge passes will be required to compute the final result of the projection query?
v. What is the I/O cost of these additional passes?

4. Consider the following query:

SELECT *
FROM R, S, T
WHERE R.A = S.A and R.B = T.B AND T.C = S.C
AND R.D = 16 AND S.F = 17

A. Give six algebraic expressions that :
(I) contain no cartesian product, (II) have pushed selections down, (III) each join is associated with an atomic condition (not a conjunctive condition).

B. Find the optimum plan, i.e., an algebraic expression where selections and projections are annotated with execution primitives. The execution primitives are: SCAN and INDEX for the selection operator and LOOPS, INDEX, MERGE, and HASH for join. Compute an estimation of the optimal plan's cost. Assume the following schemas and statistics for the relations.

- R(A, B, D, E) has 1000000 tuples, an index on D, V(D,R)=1000, V(A,R)=1000000, V(B,R)=1000000.
- S(A, C, F, G) has 1000000 tuples, an index on F, V(F,S)=1000, V(A,S)=1000000, and V(C,S)=1000000
- T(B, C, H, I) has 1000000 tuples, an index on B, with V(B,T)=1000000, V(C,T)=1000000.

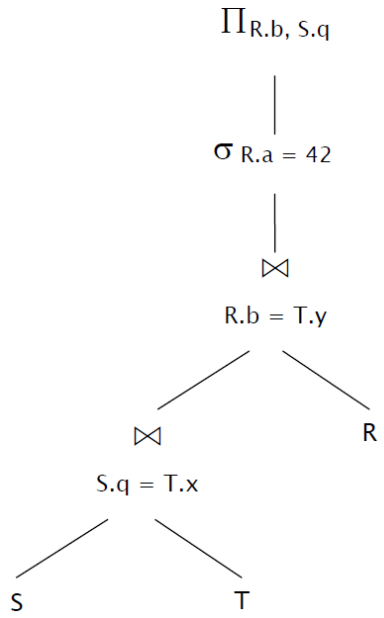
Assume the block size is 4096 Bytes, the block header size is 96 bytes, each indexed attribute occupies 5 bytes and the tuple size of each relation is 80 bytes. To estimate the size of the joins use the following information:

- For every tuple of R there is exactly one tuple of S with the same A value and one tuple of T with the same B value. Equivalently, the chances that a given S tuple joins a given R tuple are 1/1000000.
- For every tuple of S there is exactly one tuple of R with the same A value and one tuple of T with the same C value.
- For every tuple of T there is exactly one tuple of R with the same B value and one tuple of S with the same B value.

Assume there is one megabyte of main memory available. Hint: You should use the main memory to avoid copying intermediate results to the disk, if there is enough space in main memory

5. We would like to explore query plans involving three relations: R(a,b) S(p,q) T(x,y)
There are no foreign key relationships between these tables. Each table is clustered on its primary key and has a B+ tree index on that key. We have the following statistics about the tables:
B(R) = 50 T(R) = 2000 V(R,a) = 50 V(R,b) = 20
B(S) = 200 T(S) = 4000 V(S,p) = 400 V(S,q) = 100
B(T) = 100 T(T) = 1000 V(T,x) = 200 V(T,y) = 100

Now consider the following logical query plan:



A. What is the estimated cost of the logical query plan shown above?

B. Assume that we have M = 180 blocks of memory available. Without rearranging this logical plan, pick the best (cheapest) physical plan to execute it in the available memory. Annotate the diagram showing which physical operations you would use. You should indicate how you plan to access the files as well as how to implement the operations. Be sure it is clear whether intermediate results are materialized on disk or not, which relations are stored in hash tables or other data structures in memory, etc.