CS 411 Database Systems: Homework #5

Fall 2014

Released 11/20/2014, Due 12/4/2014 at 11:59pm
Late submissions are not allowed, so please plan to submit well in advance of the deadline

Problem 1 - Estimating Cost (25 points)

Consider the relations A(x,y,z), B(w,x), and C(u,v,w), with the following properties:

A(x,y,z)	B(w,x)	C(u,v,w)
T(A) = 2500 V(A,x) =30 V(A,y)=200 V(A,z)=40	T(B) = 1000 V(B,w)=250 V(B,x) = 50	T(C) = 6000 V(C,u) = 10 V(C,v)=40 V(C,w) =100

Where T(R) = number of tuples in relation R
and V(R, a) = number of distinct values of attribute a in relation R

Estimate the sizes (measured in number of tuples) of the result of the following expressions:

1. A X C
2. $A \bowtie B$
3. SELECT u FROM C WHERE u=20
4. $\sigma_{x=20 \text{ and } u=30} (B \bowtie C)$

Problem 2 - Dynamic Programming (25 points)

Consider the following relations:

A(x,y,z)	B(w,x)	C(u,v,w)	D(u,z)
T(A) = 2500 V(A,x) =30 V(A,y) =200 V(A,z) =40	T(B) = 1000 V(B,w) =250 V(B,x) = 50	T(C) = 6000 V(C,u) = 10 V(C,v) =40 V(C,w) =100	T(D) = 2000 V(D,u) = 100 V(D,z) = 50

We want to join all these relations as efficiently as possible. As we learned in class, there is a dynamic programming approach to help determine the best join order.

Determine the most efficient way to do the join. Show your work by completing the following table (each step in the dynamic programming algorithm should be one row):

Subset	Size	Lowest Cost	Lowest-cost plan
...	...	...	...

Details: when doing the calculations, you should:

- Use the “heuristic to reduce search space” discussed in class, which is to “restrict to trees without Cartesian products” when creating the dynamic programming tables. In other words, only consider joins of those relations where a natural join is feasible
- Assume that the join operation is symmetric, i.e. the plan (R1 R2) is the same as the plan (R2 R1). (In general, this need not be the case.)

Problem 3 - Query Execution (25 points)

Consider the following relations:

- Relation R has 5,000 tuples, 250 tuples per block
- Relation S has 3,500 tuples, unknown tuples per block

You know that (1) the number of blocks in memory is 41; (2) the cost of joining R and S using a nested-loop join is 200.

Answer the following questions:

1. How many tuples per block does S have? (Don't forget to show your calculations. Also, if you make any assumptions about the relative sizes of R and S, you must state *and justify* them).

2. Using your answer above, what is the cost of joining R and S using the sort-merge join algorithm discussed in class?

3. What is the cost of joining R and S using a hash-based join?

Problem 4 - Materializing and Pipelining (25 points)

Suppose we want to execute the following query:

$$(R(w,x) \bowtie S(x,y)) \bowtie U(y,z)$$

And you are given the following facts:

1. B(R) = 800, B(S) = 3,000, B(U) = 3000
2. The intermediate result R ⋈ S occupies k blocks
3. Both joins will be implemented as hash joins (either one-pass or two-pass, depending on k)
4. The memory buffer has 41 blocks

In the following table, find the missing values a, b, c, d, e, f, g, and h

k	Pipeline or Materialize?	Algorithm for final join	Total disk I/O*
0 <= k < a	Pipeline	One-pass	f
a <= k < b	Materialize	d buckets, two-pass	g
b <= k < c	Materialize	e buckets, two-pass	h

* Per the convention in class, don't include the I/O necessary to write the final answer to disk.