# Physical Operators

Exercises

2015-2016

**Exercise 1.** Consider the join $R \bowtie_{R.A=S.B} S$ and the following information on $R$ and $S$.

- Relation $R$ contains 10000 tuples and has 10 tuples per block.
- Relation $S$ contains 2000 tuples and has 10 tuples per block.
- Attribute $B$ of relation $S$ is the primary key for $S$.
- Neither $R$ nor $S$ is sorted on the join attribute.
- Neither relation has any indexes built on it.
- There are 52 main memory buffers available.

1. What is the cost (in disk I/O's) of joining $R$ and $S$ using the tuple-based nested loop join? What is the minimum number of buffer pages required for this cost to remain unchanged?
2. What is the cost (in disk I/O's) of joining $R$ and $S$ using the block-based nested loop join? What is the minimum number of buffer pages required for this cost to remain unchanged?
3. What is the cost (in disk I/O's) of joining $R$ and $S$ using a sort-merge join? What is the minimum number of buffer pages required for this cost to remain unchanged?
4. What is the cost (in disk I/O's) of joining $R$ and $S$ using a hash join? What is the minimum number of buffer pages required for this cost to remain unchanged?
5. What join algorithm yields the least cost if you were free to choose the number of free buffers? Briefly motivate your answer and give the exact optimal cost.
6. How many tuples does the join of $R$ and $S$ produce, at most, and how many blocks are required to store the result of the join back on disk?

7. Would your answer to any of the previous questions in this exercise change if you were told that $R.A$ is a foreign key that refers to $S.B$?

**Exercise 2.** Consider again the join of $R$ and $S$ from the previous exercise.

1. Assume that *unclustered* BTree indexes exist on $R.A$ and $S.B$ and that there are 15 buffers available.
   (a) Is an index nested loop join using either one of these indexes a cheaper alternative for performing the join than a block-based nested loop join? Explain
   (b) Would your answer change if only 5 buffers are available?
   (c) Would your answer change if $S$ contained only 10 tuples instead of 2000 tuples?
2. Next assume that instead *clustered* BTree indexes exist on $R.A$ and $S.B$. Again, 15 buffers are available
   (a) Is an index nested loop join using either one of these indexes a cheaper alternative for performing the join than a block-based nested loop join? Explain
   (b) Would your answer change if only 5 buffers are available?
   (c) Would your answer change if $S$ contained only 10 tuples instead of 2000 tuples?
3. If only 15 buffers were available, what would be the cost of a sort-merge join? What would be the cost of a hash join?
4. If the size of $S$ were increased to also be 10000 tuples, but only 15 buffers were available, what would be the cost of a sort-merge join? What would be the cost of a hash join?
5. If the size of $S$ were increased to also be 10000 tuples, and 52 buffers were available, what would be the cost of sort-merge join? What would be the cost of hash join?