

## Quiz 9: Query execution (10 points), 10 minutes

Consider natural-joining two relations  $R(A, B)$  and  $S(A, C)$  using the **simple sort-based join** algorithm. Suppose  $M = 101$  pages,  $B(R) = 5,000$  blocks, and  $B(S) = 20,000$  blocks. Assume sorting only uses 100 pages and the situation where we have too many joining tuples does not arise.

1. [6 points] Describe the steps of the algorithm (i.e., how many passes, what are the outputs generated at each pass, and the size of the outputs).

**Sort-Merge Step**For R

*pass0 (sort): use 100 buffers to sort R, generate 50 runs of size 100 blocks.*

*pass1 (merge): merge 50 runs into 1 sorted run of size 5000 blocks.*

For S

*pass0 (sort): use 100 buffers to sort S, generate 200 runs of size 100 blocks.*

*pass1 (merge): merge 200 runs into 2 sorted runs of size 10000 blocks.*

*Pass2 (merge): merge 2 runs into 1 sorted run of size 20000 blocks.*

**Join Step**

*Do a 2-way, one from R and one from S, merge on the attribute 'A'.*

2. [2 points] What is the total cost (i.e., the number of block I/O's) of the algorithm?

$$\text{Total cost} = 4B(R) + 6B(S) + B(R) + B(S) = 165000 \text{ block I/Os}$$

3. [2 points] Explain the main difference between the **sort-merge** join and **simple sort-based** join algorithm.

*The main difference is that we completely sort the relations (1 run for each) before we do the join in **simple sort-based** join. However, in **sort-merge** join, we start the join as soon as the total runs of two relations can fit in the memory buffers.*