

Question 1: Parallel Sort-Merge Join [240 points]

- (i) [10 points] **Parallel Join Algorithms:**
Explain how to parallelize a sort-merge join.
- (ii) [10 points] **SIMD:**
Distinguish between: SIMD and SISD.
- (iii) [10 points] **SIMD:**
List a benefit and a drawback of SIMD.
- (iv) [10 points] **Parallel Sort-Merge Join:**
List the phases of a sort-merge join.
- (v) [10 points] **Parallel Sort-Merge Join:**
Explain why merge sort is used for sorting data instead of quicksort.
- (vi) [10 points] **Partition Phase:**
Why is the partition phase optional?
- (vii) [10 points] **Partition Phase:**
List two types of partitioning. When is more advantageous to perform a sort-merge join?
- (viii) [10 points] **Sort Phase:**
Explain why quicksort is good enough in a disk-centric DBMS.
- (ix) [10 points] **Sort Phase:**
List the levels in cache-conscious sorting.
- (x) [10 points] **Sort Phase:**
Define a sorting network.
- (xi) [10 points] **Sort Phase:**
Distinguish between a bitonic merge network and a sorting network.
- (xii) [10 points] **Sort Phase:**
Define multi-way merging. Explain the purpose of using a cache-sized FIFO queue.
- (xiii) [10 points] **Merge Phase:**
Explain when backtracking is done during the merge phase.
- (xiv) [10 points] **Merge Phase:**
List the types of sort-merge join.
- (xv) [10 points] **Merge Phase:**
Distinguish between: (1) Multi-Way, (2) Multi-Pass, and (3) Massively Parallel Sort-Merge algorithms.
- (xvi) [10 points] **Merge Phase:**
When can the hardware prefetcher mask the latency penalty of going over NUMA regions?
- (xvii) [10 points] **Merge Phase:**
List the rules for parallelization.

- (xviii) **[10 points] Merge Phase:**
Distinguish between: (1) merge sort, (2) quick sort, and (3) heap sort.
- (xix) **[10 points] Evaluation:**
Explain why the multi-way sort-merge join algorithm outperforms other algorithms.
- (xx) **[10 points] Evaluation:**
Explain why the massively parallel sort-merge join algorithm does not work well in practice.
- (xxi) **[10 points] Evaluation:**
Compare the performance of sort-merge join and hash join algorithms.
- (xxii) **[10 points] Evaluation:**
When is sort-merge join useful as opposed to hash join?
- (xxiii) **[10 points] Evaluation:**
Which DBMS component chooses between sort-merge join and hash join algorithms?
- (xxiv) **[10 points] Evaluation:**
How does the optimizer choose between sort-merge join and hash join algorithms?