2.3 ALGORITHMS · 2.3.1(F)

Merge & quick sort

The fast divide-and-conquer sorts. Spec 2.3.1(f).

01 Merge sort

Divide	Split in half until single items.
Conquer	**Merge** sorted sub-lists back together.
Complexity	O(n log n) — always.
Memory	**Not in place** (needs extra lists).

02 Quick sort

Pivot	Choose a pivot value.
Partition	Smaller left, larger right; pivot in place.
Recurse	Quick sort each part.
Complexity	O(n log n) avg; **O(n^2) worst** (poor pivots).

03 Comparison

Merge	Guaranteed O(n log n); needs **extra memory**; divides by **position**.
Quick	Usually fastest; **in place**; divides by **value**; risky worst case.
vs simple	Both beat bubble/insertion (O(n^2)) on large data.
Both are	**divide & conquer** → O(n log n).

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on merge/quick sort.

01

Merge splits by **position**; quick splits by **value** (pivot).

02

Both are **O(n log n)** — far better than O(n^2) sorts on big data.

03

Merge sort is **not in place** (extra memory).

04

Quick sort can be **O(n^2)** with consistently poor pivots.

05

Both are **divide & conquer**: split into same-type sub-problems, combine.

06

Trace merge with a **split/merge tree**; quick by the **partitions**.