Interview-Favorite Sorting Algorithms (Learn with Confidence)

✅ Sorting Algorithms That Actually Matter

🔹 1. QuickSort

- **Why learn?** It's the **default go-to in interviews** for sorting — fast, efficient, and great average-case time.
- **Use in real-world?** Yes, behind the scenes of many built-in sorts (like in C++, Java, Python).
- **Time Complexity:**
  - Average: O(n log n)
  - Worst: O(n²) — but with randomized pivot, this is rare.

🔹 2. MergeSort

- **Why learn?** It's **stable** and always gives O(n log n) — very useful when data stability matters.
- **Use in real-world?** Used in **LinkedList sorting** and **multi-threaded** or **external sorting**.
- **Time Complexity:** Always O(n log n)

🔹 3. HeapSort

- **Why learn?** Helps build **Heap data structure** understanding + used in **priority queues**.
- **Use in real-world?** Rarely used as main sort, but its core concept (heap) is widely used.
- **Time Complexity:** O(n log n)

Algorithm	Must for Interviews?	Use in Production?	Notes
QuickSort	✅ Yes	✅ Yes (internally)	Know logic, partition, and dry run
MergeSort	✅ Yes	✅ Yes	Especially when stability is key
HeapSort	✅ Good to Know	👎 Rarely used	Learn heap separately too
CountingSort	🟡 Sometimes Asked	✅ For small integers	Not general purpose
Radix/BucketSort	🟡 Advanced Use	✅ Specific use-cases	Good for numbers/digits only
Bubble/Selection/Insertion	👎 Not Asked	👎 Not used	Only for basics/warmups

---

🧠 So What Should You Focus On?

✅ Learn **QuickSort**, **MergeSort**, and understand **HeapSort**
✅ Practice sorting-related LeetCode/DSA problems
✅ Use built-in sorting methods in projects (e.g., `Arrays.sort()` in Java = tuned QuickSort for primitives, MergeSort for objects)

🧪 Final Verdict

Focus your energy on:

Goal	Sorting To Master
Crack Interviews	✅ QuickSort, ✅ MergeSort, 🟡 HeapSort
Build Projects	Use built-in sort methods