**Additional Notes:**

* **Tim sort** is usually the best choice in practice for general-purpose sorting because of its *O* (*n* log *n*) complexity and stability, which is why it’s used in many standard libraries.
* **Merge Sort** and **Heap Sort** are strong contenders for their guaranteed 𝑂 (𝑛 log 𝑛) performance, though Merge Sort has the advantage of stability.
* **Quicksort** is often the fastest in practice on average, but its worst-case performance can be poor without good pivot selection strategies.
* **Insertion Sort**, **Bubble Sort**, and **Selection Sort** are typically used for educational purposes or very small datasets. Among these, Insertion Sort is usually the most efficient.

**Choosing the Right Algorithm:**

* **Small datasets**: Insertion Sort or Bubble Sort (if data is nearly sorted).
* **Large datasets with strict space constraints**: Heap Sort.
* **General-purpose sorting in practical applications**: Tim sort or Quicksort with a good pivot strategy.
* **When stability is required**: Merge Sort or Tim sort.