Sorting Algorithms Comparison Table

Algorithm	Best Case	Average Case	Worst Case	Space	Stable?	In-Place?	Type	Whe
Insertion Sort	O(n)	$O(n^2)$	$O(n^2)$	O(1)	Yes	Yes	Comparison-based	Small or nearly s
Merge Sort	O(n log n)	O(n log n)	O(n log n)	O(n)	Yes	No	Divide & Conquer	Large datasets w
Quick Sort	O(n log n)	O(n log n)	$O(n^2)$	O(log n)	No	Yes	Divide & Conquer	Fast general-purp
Heap Sort	O(n log n)	O(n log n)	O(n log n)	O(1)	No	Yes	Comparison-based	When you need g
Counting Sort	O(n + k)	O(n + k)	O(n + k)	O(k)	Yes	No	Non-comparison-based	Integers or categ
Radix Sort	O(nk)	O(nk)	O(nk)	O(n + k)	Yes	No	Non-comparison-based	Large number of
Bucket Sort	O(n + k)	O(n + k)	$O(n^2)$	O(n + k)	Yes*	No	Non-comparison-based	Uniformly distribu

Notes:

- Stable: Preserves order of equal elements.

- In-Place: Uses constant extra space.

- k = range (Counting), number of digits (Radix), or buckets (Bucket).

- Bucket Sort is stable only if the inner sort is stable (e.g., Insertion Sort).