

Empirical analysis of sorting algorithms

AHMAD SABBAH 202110725

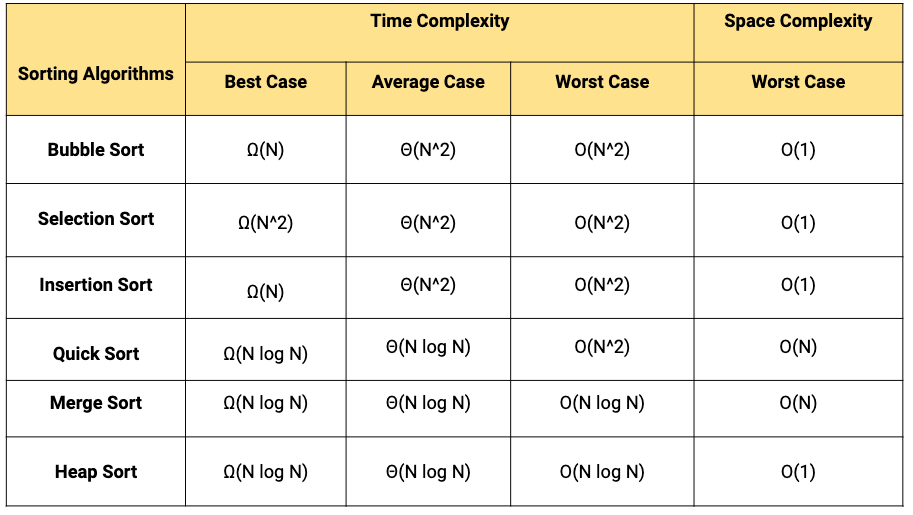


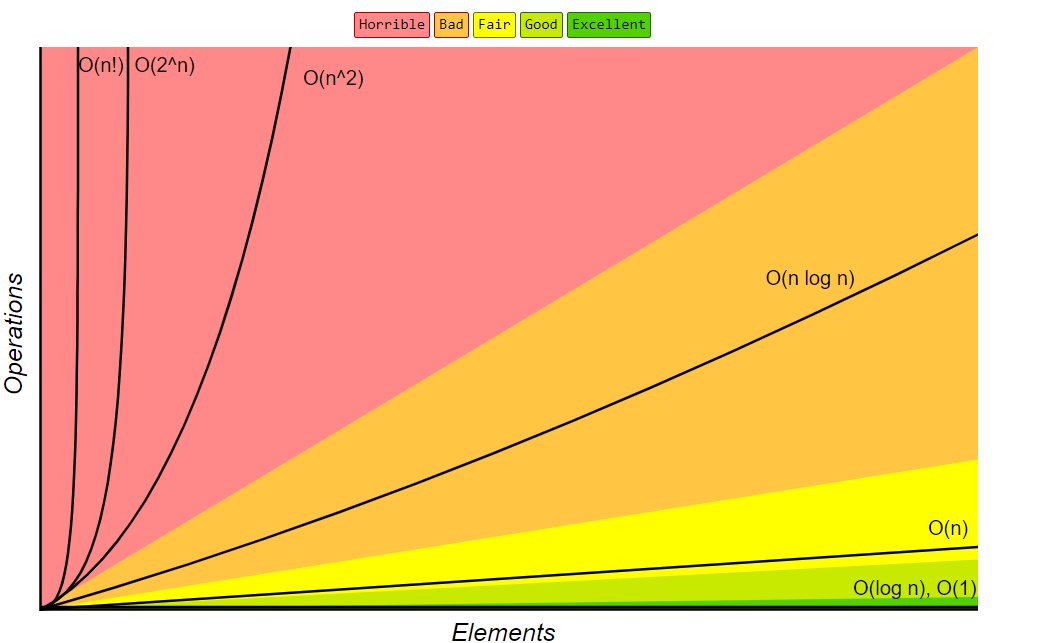
January 25, 2024

AAUP

Time Complexity & Space Complexity

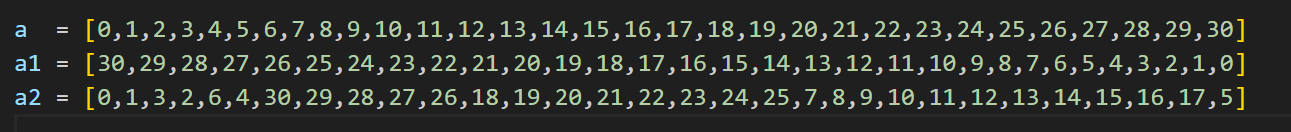
For Sorting Algorithms





Experimenting with some test cases and monitoring their respective times

Top of Form



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quick Sort | Merge Sort | SelectionSort | Bubble Sort | **Test case** |
| 20451501  nanoseconds | 11138200 nanoseconds | 10203200  nanoseconds | 0  nanoseconds | **a** |
| 40738600 nanoseconds | 21548800 nanoseconds | 11401200 nanoseconds | 6403400  nanoseconds | **a1** |
| 30747600  nanoseconds | 17282100 nanoseconds | 10309800 nanoseconds | 8001203  nanoseconds | **a2** |

These values ​​are the arithmetic **average** of several times of **running** the program

Code : https://github.com/as-ahmadsabbah/AlgorithmsCourse/tree/main/assigment%20sort

Observing the performance of the more time-consuming sorting system Quick Sort than the others seems to contradict the scientific facts. The order you mentioned, with Quick Sort being the slowest followed by Merge Sort, Selection Sort, and Bubble Sort, is due to the specific computer hardware and processor used for testing

However, the typical order of algorithmic complexity is as follows

Slow to fast.

Bubble Sort

Selection Sort

Quick sort

Merge Sort

This difference in performance may be largely due to the nature of the computer system used. It is important to note that the efficiency of serialization algorithms can vary depending on factors such as data size, priority configuration, and hardware architecture. In theory, Quick Sort and Merge Sort are generally expected to outperform Bubble Sort and Selection Sort for large data sets.

STABILITY OF SORTING ALGORITHMS

Quick Sort

Stability: Quick Sort is commonly not stable. During the partitioning step, the relative order of equal elements won't be preserved. This is because the set of rules does no longer assure the order of identical elements throughout the partitioning technique.

Merge Sort

Stability: Merge Sort is a solid sorting algorithm. It keeps the relative order of same factors within the input array. This stability is performed throughout the merging step, in which equal factors from the two halves are merged in a way that their authentic order is preserved.

Selection Sort

Stability: Selection Sort is typically no longer stable. During each new release, the set of rules selects the minimum element and swaps it with the detail at the modern-day function. This swapping operation may additionally disrupt the relative order of equal factors.

Bubble Sort

Stability: Bubble Sort is a solid sorting algorithm. It compares adjacent elements and swaps them if they're inside the incorrect order, but it handiest plays swaps while adjacent factors aren't identical. This ensures that the relative order of identical elements is maintained

In summary

Merge Sort and Bubble Sort are strong sorting algorithms.

Quick Sort and Selection Sort are commonly no longer solid, as they'll interchange the positions of same factors.