

Bubble Sort:

Bubble Sort is a simple comparison-based sorting algorithm. It works by repeatedly swapping adjacent elements if they are in the wrong order. The algorithm iterates through the list multiple times until the entire list is sorted. During each iteration, the largest (or smallest, depending on the sorting order) element "bubbles" up to its correct position.

The key steps of the Bubble Sort algorithm are as follows:

- 1- Compare the first pair of adjacent elements.
- 2- If they are in the wrong order, swap them.
- 3- Move to the next pair of adjacent elements and repeat step 2.
- 4- Continue this process until the list is sorted.

Bubble Sort has a time complexity of $O(n^2)$ in the average and worst cases, where "n" represents the number of elements being sorted. It is not considered an efficient sorting algorithm for large datasets.

Merge Sort:

Merge Sort is a divide-and-conquer algorithm that breaks down the sorting process into smaller subproblems. It recursively divides the input list into two halves, sorts them independently, and then merges them back into a single sorted list. The merging step compares elements from the two sorted sublists and combines them in the correct order.

The key steps of the Merge Sort algorithm are as follows:

- 1- Divide the unsorted list into two halves.
- 2- Recursively sort each half independently using Merge Sort.
- 3- Merge the sorted halves by comparing elements and placing them in the correct order.

Merge Sort has a time complexity of $O(n \log n)$ in the average and worst cases. It is considered to be more efficient than Bubble Sort for larger datasets, as its time complexity grows at a slower rate.

The main differences between Bubble Sort and Merge Sort are:

- 1- Approach: Bubble Sort compares adjacent elements and gradually moves the largest/smallest elements to their correct positions. Merge Sort, on the other hand, divides the list into smaller sublists, sorts them independently, and merges them back together.
- 2- Time Complexity: Bubble Sort has a time complexity of $O(n^2)$, while Merge Sort has a time complexity of $O(n \log n)$. Merge Sort is generally more efficient, especially for larger datasets.
- 3- Stability: Bubble Sort is a stable sorting algorithm, which means that the relative order of equal elements is preserved. Merge Sort is also stable, as it maintains the relative order of elements during the merging step.
- 4- Space Complexity: Bubble Sort has a space complexity of $O(1)$ since it only requires a constant amount of additional space. Merge Sort has a space complexity of $O(n)$ since it needs additional space to store the sorted sublists during the merging step.