Merge Sort is a divide-and-conquer algorithm that divides the input list into smaller sublists, sorts them recursively, and then merges the sorted sublists to produce a sorted output list.

It takes advantage of the fact that it is easier to merge two sorted lists than to sort a single unsorted list.

### Here's how the Merge Sort algorithm works:

- 1. Divide the unsorted list into two equal-sized sublists (or approximately equal-sized for odd-length lists).
- 2. Recursively sort the two sublists by applying the Merge Sort algorithm to each sublist.
- 3. Merge the two sorted sublists back into a single sorted list.
- 4. Repeat steps 2-3 until the entire list is sorted.

# The key step in the Merge Sort algorithm is the merging of two sorted sublists.

To merge the sublists, follow these steps:

- 1. Create an empty auxiliary array to hold the merged elements.
- 2. Compare the first elements of both sublists. Take the smaller (or larger) element and append it to the auxiliary array.
- 3. Move the pointer of the sublist from which the selected element was taken to the next element.
- 4. Repeat steps 2-3 until one sublist is fully processed.
- 5. Append the remaining elements of the other sublist to the auxiliary array.
- 6. Copy the elements from the auxiliary array back into the original list, replacing the unsorted elements with the sorted elements.

## Example:

Consider the following unsorted list of integers: [5, 3, 8, 2, 1].

#### Step 1:

• Divide the list into two sublists: [5, 3] and [8, 2, 1].

#### Step 2:

- Recursively sort the first sublist [5, 3]:
  - Divide the sublist into [5] and [3].
  - Since each sublist contains only one element, they are already considered sorted.
- Recursively sort the second sublist [8, 2, 1]:
  - Divide the sublist into [8] and [2, 1].
  - Recursively sort the second sublist [2, 1]:
    - Divide the sublist into [2] and [1].
    - Since each sublist contains only one element, they are already considered sorted.
    - Merge the sublists [2] and [1]. The merged sublist becomes [1, 2].
  - Merge the sublists [8] and [1, 2]. The merged sublist becomes [1, 2, 8].

#### Step 3:

• Merge the two sorted sublists [5, 3] and [1, 2, 8]. The merged sublist becomes [1, 2, 3, 5, 8].

The list is now sorted in ascending order: [1, 2, 3, 5, 8].

Merge Sort has a time complexity of O(n log n) in all cases, where 'n' is the number of elements in the list.

Merge Sort is considered efficient for large lists and guarantees a consistent performance regardless of the initial ordering of the elements.