**The Merge Sort algorithm**

is a popular and efficient sorting algorithm that follows the divide-and-conquer approach. It works by dividing the unsorted list into smaller sublists, sorting them recursively, and then merging them back together to obtain the final sorted list.

Here's a brief explanation of how the Merge Sort algorithm works:

1. Divide: The unsorted list is divided into two halves repeatedly until each sublist contains only one element or is empty.
2. Conquer: The sublists are recursively sorted. This is done by applying the Merge Sort algorithm to each sublist, dividing it further if necessary.
3. Merge: The sorted sublists are merged back together to obtain a single sorted list. This merging process compares the elements of the sublists and places them in the correct order.
4. Repeat: Steps 2 and 3 are repeated until the entire list is sorted.

The key step in the Merge Sort algorithm is the merging process, where two sorted sublists are combined into a single sorted list. This is achieved by comparing the elements from both sublists and arranging them in the correct order. The merging process continues until all elements from both sublists have been merged.

Merge Sort has a time complexity of O(n log n), which means it performs well even for large lists. It is a stable sorting algorithm, which means it preserves the relative order of equal elements during the sorting process. However, it requires additional space for the merging process, making it less memory-efficient compared to some other sorting algorithms.

Overall, Merge Sort is widely used due to its efficiency and stability, especially when sorting large data sets or when stability is a crucial requirement.

**Bubble sort algorithm** is a simple sorting algorithm that repeatedly steps through a list of elements, compares adjacent elements, and swaps them if they are in the wrong order. The algorithm gets its name from the way smaller elements "bubble" to the top of the list during each iteration.

Here's a brief explanation of how the Bubble sort algorithm works:

1. Start with an unsorted list of elements.
2. Compare the first element with the second element. If the first element is larger, swap the two.
3. Move to the next pair of elements (second and third) and compare them. Again, swap if they are in the wrong order.
4. Continue this process, comparing and swapping adjacent elements, until you reach the end of the list.
5. At this point, the largest element in the list will be in its correct position at the end.
6. Repeat steps 2-5 for the remaining unsorted portion of the list (excluding the last element).
7. Keep repeating the process until the entire list is sorted.

The bubble sort algorithm has a time complexity of O(n^2), which means that it is not very efficient for large lists. However, it is easy to understand and implement, making it useful for educational purposes or for sorting small lists where efficiency is not a major concern.

It's worth noting that there are more efficient sorting algorithms available, such as quicksort or mergesort, which have better average and worst-case time complexities.