**Bubble Sort:**

Bubble Sort is a simple sorting algorithm that works by repeatedly swapping adjacent elements if they are in the wrong order. It gets its name from the way smaller elements “bubble” to the top of the list.

Here’s how it works:

1. Starting from the first element of the list, compare the current element with the next element.
2. If the current element is greater than the next element, swap them.
3. Move to the next pair of elements and repeat step 2 until the end of the list is reached.
4. Repeat steps 1-3 until no more swaps are needed.

Bubble Sort has a worst-case and average-case time complexity of O(n^2), where n is the number of items being sorted. This makes it inefficient for large lists and it is not commonly used in practice for sorting large data sets.

**Merge Sort:**

Merge sort is a divide-and-conquer algorithm that works by dividing an array into two halves, sorting each half separately, and then merging the sorted halves back together. The key to the merge sort algorithm is the merge step, which combines two sorted sub-arrays into a single sorted array.

Here’s how the merge sort algorithm works:

1. Divide the unsorted list into n sub-lists, each containing one element (a list of one element is considered sorted).
2. Repeatedly merge sub-lists to produce new sorted sub-lists until there is only one sub-list remaining. This will be the sorted list

**Selection Sort:**

Selection sort is a simple sorting algorithm that works by repeatedly finding the minimum element from an unsorted portion of the list and moving it to the beginning of the list. The algorithm maintains two sub-lists in a given list:

1. The sorted sub-list.
2. The unsorted sub-list.

Initially, the sorted sub-list is empty and the unsorted sub-list is the entire input list. The algorithm proceeds by finding the smallest element in the unsorted sub-list and swapping it with the leftmost unsorted element, moving that element into sorted sub-list.