Bubble sort: Bubble sort is one of the simplest sorting algorithms. The two adjacent elements of a list are checked and swapped if they are in wrong order and this process is repeated until we get a sorted list. The steps of performing a bubble sort are: Compare the first and the second element of the list and swap them if they are in wrong order. Compare the second and the third element of the list and swap them if they are in wrong order. Proceed till the last element of the list in a similar fashion. Repeat all of the above steps until the list is sorted.

Selection Sort is similar to the hand picking where we take the smallest element and put it in the first position and the second smallest at the second position and so on. We first check for smallest element in the list and swap it with the first element of the list. Again, we check for the smallest number in a sublist, excluding the first element of the list as it is where it should be (at the first position) and put it in the second position of the list. We continue repeating this process until the list gets sorted. We follow the following steps to perform selection sort: Start from the first element in the list and search for the smallest element in the list. Swap the first element with the smallest element of the list. Take a sublist (excluding the first element of the list as it is at its place) and search for the smallest number in the sublist (second smallest number of the entire list) and swap it with the first element of the list (second element of the entire list). Repeat the steps 2 and 3 with new sublists until the list gets sorted.

QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways. Always pick first element as pivot. Always pick last element as pivot (implemented below)Pick a random element as pivot. Pick median as pivot. The key process in quickSort is partition(). Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time.

Merge sort based on Divide and Conquer approach. That means it divides the array into two parts, those sub-arrays will be divided into other two equal parts. We will divide the array in this manner until we get single element in each part because single element is already sorted. After completion of dividing array, its time to conquer or merge them together but in sorted manner. Hence we get a sorted array. Merge sort is recursive(method that call itself).Very efficient for large data set.