\*Sorting\*

# Sorting Algorithms

[Online resource for codes.](https://www.cs.cmu.edu/~rdriley/121/notes/sorting.html)

Five techniques have been covered for sorting, namely,

## Bubble Sort:

In this technique we compare adjacent pairs and swap in the desired order. Repeating this process till end of the loop drags the largest number (when sorting in ascending order) to the end of the array. Thus, last element is fixed; now the next loop is run till the last non-fixed element only.

## Selection Sort:

In this technique we fix one element at a time from left to right. Outer loop selects the index to be fixed; inner loop finds the smallest element in the unsorted part to e fixed at the selected index.

## Insertion Sort:

This technique divides the array into two parts. Outer loop selects the next element in the un-ordered part; inner loop inserts the selected element in at its correct position in the former ordered part.

## Merge Sort:

This technique uses divide and conquer. The array is split into two parts at the mid recursively till only one element remains. Then a merge function is called that merges the adjacent arrays in sorted order. When all the split parts are merged the array shall be sorted in the desired order.

## Quick Sort:

This technique uses divide and conquer but unlike merge sort doesn’t us O(n) space. A pivot index is selected and placed at its correct place as in a sorted array. Pivot is placed such that all elements on left are smaller and all on right are larger than the pivot element. This is done recursively on either of the parts fixing the pivot element in each recursion.

### Summary of sorting efficiency:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sort** | Best Case | Average Case | Worst Case | Stable? | Adaptive? |
| Bubble | Ω (N) | Θ (N2) | O(N2) | Y | Y |
| Insertion | Ω (N) | Θ (N2) | O(N2) | Y | Y |
| Selection | Ω (N2) | Θ (N2) | O(N2) | N | N |
| Merge | Ω (N log N) | Θ (N log N) | O(N log N) | Y | N |
| Quick | Ω (N log N) | Θ (N log N) | O(N2) | N | Y |

# Other Related topics

## Online Sorting:

An online sorting algorithm is one that will work if the elements to be sorted are provided one at a time with the understanding that the algorithm must keep the sequence sorted as more and more elements are added in. *Insertion* sort is online.

## Stable Sort:

A stable sort is one where, after sorting, the relative position of equal elements remains the same. *Bubble, insertion and merge* sort are stable sorts. Selection and quick sort are not stable sorts.

## Adaptive Sort:

The sorting algorithms in which the order/pre-sorting of elements improves the time complexity of the sorting algorithm is known as an adaptive sorting algorithm. In the case of adaptive sorting algorithms, if the array is already sorted it takes linear time to perform sorting i.e., O(N) time.

*Bubble, insertion and Quick* sort algorithms are adaptive.

## External Sort:

An external sorting algorithm is one where the goal is to sort data, typically provided in advance, that is so large that it cannot fit into main memory. While external sorting algorithms typically don't keep all the data to be sorted in memory at once, they usually assume that they can load any data that they need into memory at any time. *Merge* sort is external sort algorithm.