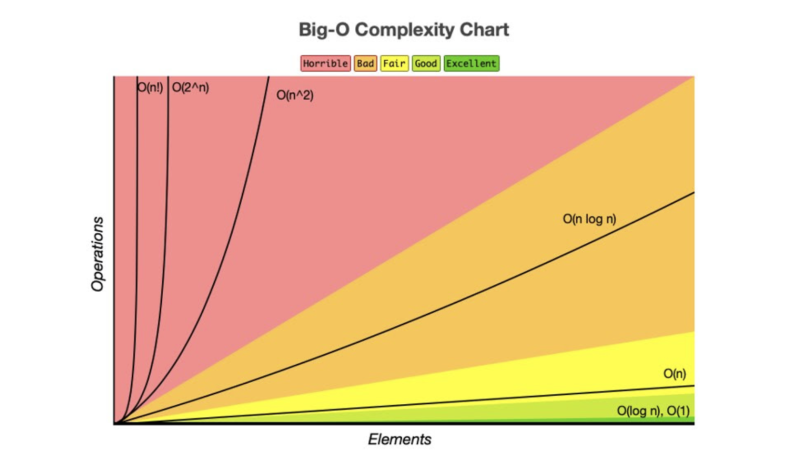
Sorting Algorithms

# The Big O Notation

Big O Notation is a mathematical concept used to describe the upper bound of the growth rate of a function as its input size increases. In computer science, Big O notation is used to classify algorithms based on how their runtime or space requirements grow with input size.

For example, an algorithm with a time complexity of O(n) is considered linear, meaning that the time it takes to complete scales linearly with the size of the input (such as the length of an array).

An algorithm with O(1) complexity is constant time, which means it takes the same amount of time to execute regardless of the input size.



# Bubble Sort

Bubble Sort is the simplest algorithm.  
It works by comparing the current value to the next and swapping them if they are in the wrong order.

For example; when sorting from lowest to highest, index[i] is compared to index[i+1] and if index[i] is higher than index[i+1] those two are swapped.  
This is repeated until the algorithm passes through the list without swapping a set of numbers.

Best case – O(n) for comparisons and O(1) for swaps

Average Case - O(n2) for comparisons and swaps

Worst Case - O(n2) for comparisons and swaps

# Mergesort

Merge sort is very popular, its efficient and stable.

Merge sort utilises the “Divide and Conquer” approach.

# Quicksort

# Cycle Sort