<https://www.geeksforgeeks.org/sorting-algorithms/?ref=shm>

**Sorting Algorithms**

* Last Updated : 14 Sep, 2022

 Read

 Discuss

A Sorting Algorithm is used to rearrange a given array or list of elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of elements in the respective data structure.

**For Example:** The below list of characters is sorted in increasing order of their ASCII values. That is, the character with a lesser ASCII value will be placed first than the character with a higher ASCII value.

# Time Complexities of all Sorting Algorithms

* **Difficulty Level :** [Easy](https://www.geeksforgeeks.org/easy/)
* **Last Updated :** 22 Sep, 2022

 Read

 Discuss

The efficiency of an algorithm depends on two parameters:

1. Time Complexity
2. Space Complexity

**Time Complexity:** Time Complexity is defined as the number of times a particular instruction set is executed rather than the total time taken. It is because the total time took also depends on some external factors like the compiler used, processor’s speed, etc.

**Space Complexity:** Space Complexity is the total memory space required by the program for its execution.

Both are calculated as the function of input size(n).

One important thing here is that in spite of these parameters the efficiency of an algorithm also depends upon the **nature** and **size of** the **input.**

**Types Of Time Complexity :**

1. **Best Time Complexity:** Define the input for which algorithm takes less time or minimum time. In the best case calculate the lower bound of an algorithm. Example: In the linear search when search data is present at the first location of large data then the best case occurs.
2. **Average Time Complexity:** In the average case take all random inputs and calculate the computation time for all inputs.  
   And then we divide it by the total number of inputs.
3. **Worst Time Complexity:** Define the input for which algorithm takes a long time or maximum time. In the worst calculate the upper bound of an algorithm. Example: In the linear search when search data is present at the last location of large data then the worst case occurs.

[](https://practice.geeksforgeeks.org/courses/complete-interview-preparation?utm_source=article&utm_medium=article&utm_campaign=complete-interview-preparation)

Following is a quick revision sheet that you may refer to at the last minute

| **Algorithm** | **Time Complexity** | | | **Space Complexity** |
| --- | --- | --- | --- | --- |
|  | **Best** | **Average** | **Worst** | **Worst** |
| [Selection Sort](http://geeksquiz.com/selection-sort/) | Ω(n^2) | θ(n^2) | O(n^2) | O(1) |
| [Bubble Sort](http://geeksquiz.com/bubble-sort/) | Ω(n) | θ(n^2) | O(n^2) | O(1) |
| [Insertion Sort](http://geeksquiz.com/insertion-sort/) | Ω(n) | θ(n^2) | O(n^2) | O(1) |
| [Heap Sort](http://geeksquiz.com/heap-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) | O(1) |
| [Quick Sort](http://geeksquiz.com/quick-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n^2) | O(n) |
| [Merge Sort](http://geeksquiz.com/merge-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) | O(n) |
| [Bucket Sort](https://www.geeksforgeeks.org/bucket-sort-2/) | Ω(n +k) | θ(n +k) | O(n^2) | O(n) |
| [Radix Sort](https://www.geeksforgeeks.org/radix-sort/) | Ω(nk) | θ(nk) | O(nk) | O(n + k) |
| [Count Sort](https://www.geeksforgeeks.org/counting-sort/) | Ω(n +k) | θ(n +k) | O(n +k) | O(k) |
| [Shell Sort](https://www.geeksforgeeks.org/shellsort/) | Ω(n log(n)) | θ(n log(n)) | O(n^2) | O(1) |
| [Tim Sort](https://www.geeksforgeeks.org/timsort/) | Ω(n) | θ(n log(n)) | O(n log (n)) | O(n) |
| [Tree Sort](https://www.geeksforgeeks.org/tree-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n^2) | O(n) |
| [Cube Sort](https://www.geeksforgeeks.org/sort-the-array-according-to-their-cubes-of-each-element/) | Ω(n) | θ(n log(n)) | O(n log(n)) | O(n) |