**Big O Notation**

<https://www.bigocheatsheet.com>

Chart

Description automatically generated

Rule Book:

Rule #1 – Worst Case

Rule #2 – Remove Constants

Rule #3 – Different terms of inputs

Rule #4 – Drop Non Dominants

String Big O

O(N \* max(K))

| **Big O Notation** | **Name** | **Example(s)** |
| --- | --- | --- |
| *O(1)* | Constant | # [Odd or Even number](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Odd-or-Even), # [Look-up table (on average)](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Look-up-table) |
| *O(log n)* | Logarithmic | # [Finding element on sorted array with **binary search**](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Binary-search) |
| *O(n)* | Linear | # [Find max element in unsorted array](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#The-largest-item-on-an-unsorted-array), # Duplicate elements in array with Hash Map |
| *O(n log n)* | Linearithmic | # [Sorting elements in array with **merge sort**](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Mergesort) |
| *O(n2)* | Quadratic | # [Duplicate elements in array \*\*(naïve)\*\*](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Has-duplicates), # [Sorting array with **bubble sort**](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Bubble-sort) |
| *O(n3)* | Cubic | # [3 variables equation solver](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Triple-nested-loops) |
| *O(2n)* | Exponential | # [Find all subsets](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Subsets-of-a-Set) |
| *O(n!)* | Factorial | # [Find all permutations of a given set/string](https://adrianmejia.com/most-popular-algorithms-time-complexity-every-programmer-should-know-free-online-tutorial-course/#Permutations) |

<https://github.com/amejiarosario/dsa.js>

Only a hash table with a perfect *hash function* will have a worst-case runtime of *O(1)*. The ideal hash function is not practical, so some collisions and workarounds lead to a worst-case runtime of *O(n)*. Still, on *average*, the lookup time is *O(1)*.