

 hello_world.py

Python Programming with Algorithmic Approach Course

Teacher



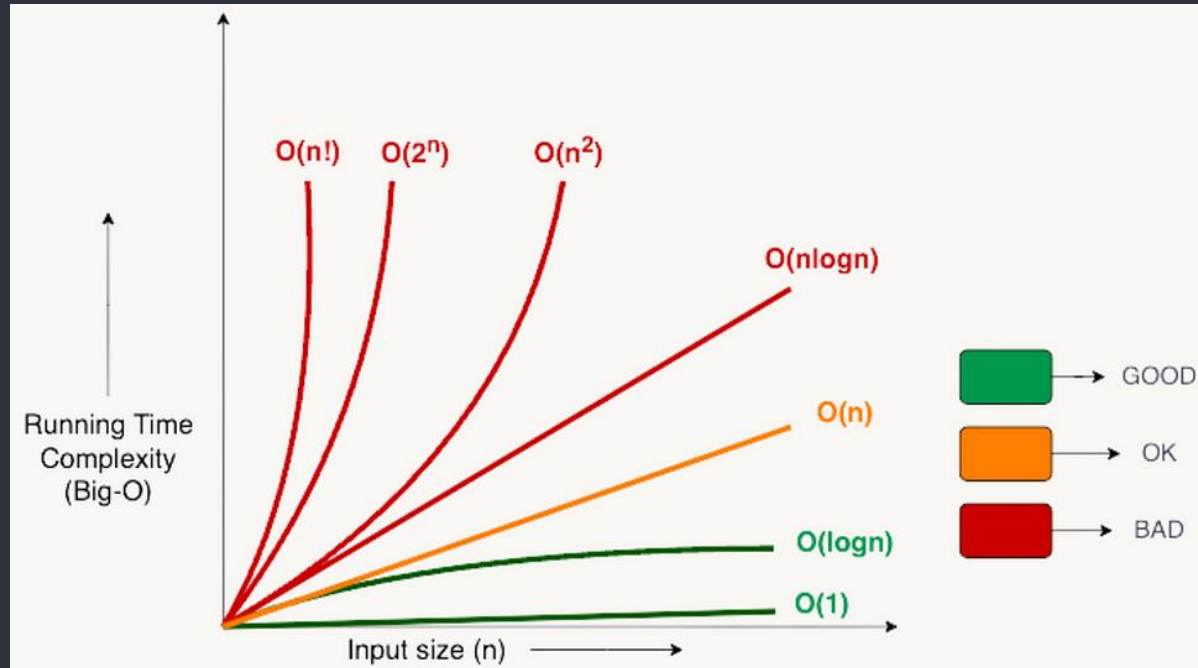
MohammadReza
Gholami

Code is poetry written
for machines!

01010000 01111001 01110100 01101000 01101111 01101110

Big O

Big O notation is a way to describe the performance or complexity of an algorithm in terms of how it scales with the size of the input. It provides an upper bound on the growth rate of an algorithm's running time in the worst-case scenario. In other words, it describes the upper limit of the function that represents the running time of an algorithm in terms of the size of its input.



$O(1)$, $O(n)$, $O(n^2)$

$O(1)$ - Constant Time

The algorithm's running time does not depend on the size of the input.

Example: Accessing an element in an array by index.

$O(n)$ - Linear Time

The running time is directly proportional to the size of the input.

Example: Finding the maximum element in an unsorted list.

$O(n^2)$ - Quadratic Time

The running time is proportional to the square of the size of the input.

Example: Bubble sort, finding all pairs of elements in an array that sum to a specific target value.



References

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

<https://medium.com/@clebertonfgc/big-o-notation-in-practice-ac192e23c16>

<https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/>

https://en.wikipedia.org/wiki/Big_O_notation

THANKS!

Do you have any questions?



+98 9939996370



mmd.gh313@gmail.com



<https://github.com/mmd00Z>



@mmd1024

