

Time and space complexity Of Algorithms



What is Algorithm

- In computer science, whenever we want to solve some computational problem then we define a set of steps that need to be followed to solve that problem. These steps are collectively known as an algorithm.

Algorithm

For example, you have two integers "a" and "b" and you want to find the sum of those two number. How will you solve this? One possible solution for the above problem can be:

- Take two integers as input
- create a variable " *sum* " to store the sum of two integers
- put the sum of those two variables in the " *sum* " variable
- return the " *sum* " variable

Properties of Good Algorithm

- **Correctness:** An algorithm is said to be correct if for every set of input it halts with the correct output.
- **Efficiency:** An efficient algorithm is always used. By the term efficiency, we mean to say that:
 1. The algorithm should efficiently use the resources available to the system.
 2. The computational time (the time taken to generate an output corresponding to a particular input) should be as less as possible.

Out of these factors, the Time and space complexity analysis deals with the efficiency of algorithms.

Time and Space Complexity

- Let's start by trying to understand what we mean by Time and space complexity

Time and space complexity are metrics used to measure the efficiency of an algorithm. Time complexity is related to the speed of an algorithm while space complexity is for the resource (memory) an algorithm consumes

You can find more in detail about time and space complexity [here](#)

Difference between Time and Space Complexity

- It's very important to know the difference between the two metrics. [Here](#) is an article that explains their difference

Time and Space Complexity

- Let's read more to understand more about this [topic](#). (This blog has questions so let's try it by ourselves before looking at the solution).

Time and Space Complexity

If you had difficulty understanding the previous resource. There are some videos and practice problems on the next slide for further clarification.

Big O



Additional Video Resource

- [Big O Notation Explained](#)

Practice Problems

- [Problems](#)

Solutions for the problems given above are provided below but, make sure that you have tried the problems by yourself before moving to the solutions.

[Solution](#)