

A Simplified Interpretation of the Time Complexity and Space Complexity

[Karuna Sehgal](#)

[Nov 20, 2017](#) · 2 min read

This blog post is a continuation of a series of blog posts about Algorithms, as it has been a hard concept for me to grasp as a programmer. Feel to check out the [first blogpost about Algorithms](#), where I provide an introduction of what Algorithms are and an example of an algorithm and the [second blog post about Data Structures](#), where I explained what are Data Structures and what are some types of Data Structures.

This blog post I want to focus on time complexity and space complexity especially before I get into Big O Notation. Big O Notation I will cover in an upcoming blog post.

Time Complexity and Space Complexity

Designing algorithms is about making decisions and decisions have tradeoffs. You might want to optimize to complete the task:

- in the shortest time
- using the least amount of space

This where complexity comes in, where you want to answer the following questions:

1. How do you know if your algorithm is good?
2. How do you compare algorithms?

So we can break this down to two concepts:

- Time Complexity -> **How long does the algorithm take**
- Space Complexity -> How much space does the algorithm use

The less complex we make the algorithm the better!

Commented [JP1]: Note 1: This is not really accurate...

Commented [JP2]: Note 2: This is not always true!

An example of an algorithm that may be more time efficient: Time Efficient Grocery Shopping

1. Drive a large car to the grocery store.
2. Buy everything that you need.

So following this algorithm saves time, but requires you to have a large car. This means lower time complexity and higher space complexity

An example of an algorithm that may be more space efficient: Space Efficient Grocery Shopping

1. Walk to the grocery store with a tote bag.
2. Buy only one item.
3. Walk home and drop off the item.
4. Repeat until done.

Following this algorithm takes a long time, but does not require a car. This means higher time complexity and lower space complexity.

Overall, time complexity and space complexity are really important factors to consider when you are designing algorithms. These decisions can have tradeoffs. Big O Notation is a language we use to describe the complexity of an algorithm. In my next blog post, I will cover Big O Notation in further detail.

Karuna Sehgal

Woman on a mission—to live the best life possible!!

Reference: Downloaded from <https://medium.com/karuna-sehgal/a-simplified-interpretation-of-the-time-complexity-and-space-complexity-78e4cddb0b19> on September, 6, 2021.