

🚩 Current Skill Introduction to Web to Algorithms

## Introduction to Algorithms

A quick search on Google will give you the following definition:

**An algorithm** is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Let's imagine together a scenario where we're in a library.

If you were to borrow a book you needed to study from the Engineering section, how would you go about doing that?

Would you:

- Pick up **every** book on the shelf until you find what you need?
- Pick up books **randomly** until you find the right one?
- Try to find it by **alphabetical order**?
- Use the library's search guide as a **reference**?

Bear in mind that these are all ways to eventually achieve the goal. Also, each method highlighted



above can be broken down into smaller steps that cumulatively achieve the task at hand. For starters, you would have to:

- Walk up to the shelf.
- Pick up the first book.
- Check if it matches what you're looking for.
- If it does, yayyyyyy!!! (that was quick).
- If it doesn't, you drop the book.
- Pick up another one and check again.



This process is continuously repeated until you find the desired book. This step by step breakdown of the whole process into actionable steps is what we call **pseudocode** in computer programming.

## Introduction to Algorithms



Let's examine the four search methods to see which is most efficient. Specifically by calculating/estimating the **time** and **energy** needed for each method. The optimal solution will be the one that consumes the least amount of time and energy.



The main properties of an algorithm which we will now consider are:

### 1. Input

An algorithm must possess 0 or more well-defined inputs supplied externally to the algorithm.

### 2. Output

An algorithm should have 1 or more well-defined outputs as desired.

### 3. Correctness

Every step of the algorithm must generate a correct output.



### 4. Definiteness

Algorithms should be clear and unambiguous. That's why every step of the algorithm should be clear and well defined.



### 5. Finiteness

The algorithm should have a finite number of steps that must be carried out to achieve the task at hand.



[< Previous](#)

[next >](#)

