# **Algorithm and Data structure in JAVA**

## **Algorithm**

“An algorithm is a series of defined instructions that perform a task”

Algorithm defines a set of rules or processes for solving a particular problem in a finite number of steps. In this all the rules and operations must be well defined and free of ambiguity. An implementation of an algorithm is usually a computer program consisting of procedures made of commands.

* Why?
  + Gives an idea of running time
  + Helps us decide on hardware requirements
  + What is feasible vs what is impossible
  + Improvement is a never ending process

## **Data structure**

A data structure is a particular way of organizing data in a computer so that it can be used effectively. For example, we can store a list of items having the same data-type using the array data structure.

## **Analysis of Algorithms**

Big-O notation allows us to compare algorithms independently of input size. Let us compare two different Algorithms by Big-O notations,

|  |  |
| --- | --- |
| Algorithm-1 | Algorithm-2 |
| The item we are searching for will be the first or last element in the list. | The item we are searching for could be anywhere within the list. |
| Best case: O(1) | Best case: O(1) |
| Worst case: O(1) | Worst case: O(n) or linear |

In deciding which algorithm to use, we often use the worst-case time complexity as a deciding factor.

Few Time complexities are,

* O(1) or constant time
* O(log(n)) or logarithmic time
* O(n) or linear time
* O(n^2) or exponential time
* O(n!) or factorial time

We can use big-O notation to describe space complexity as well. Bigg-O notation allows you to compare algorithms that perform the same task, regardless of input size.

## **String Algorithms**

Algorithms which are basically performed on strings

### Validation of a string in java

It may be like, Does a given piece of data have a certain property? Validation Algorithm basically returns Either true or false.

Example Validation of string to check its case,

Example Validation of string for at least one number one upper and lower case letter



### Normalize strings in java

Before processing a string, it can be helpful to normalize your string to a common form. Another option is to limit your input to only a certain type. We want our algorithms to be flexible and to handle different contents for this normalization can come in handy.

Example for Normalizing a string



### Basic Parsing and Searching strings in java

A common programming task is to search for a piece of data within a string.

* If you know nothing about string, you must check every character.
* If the string character are sorted, you can optimize your algorithm based on order.

Regardless of what assumptions you make, you’ll need to be able to parse a string’s contents. The least complex and least efficient way to search for a subset of characters within a string is to use the built-in contains method.



If we wanted to check each character individually, we need to loop through string



### Reverse a sting using string builder

As string are immutable if we want to edit t it would create a new string and replace it to overcome that we use string builder which dynamically adjust itself.

