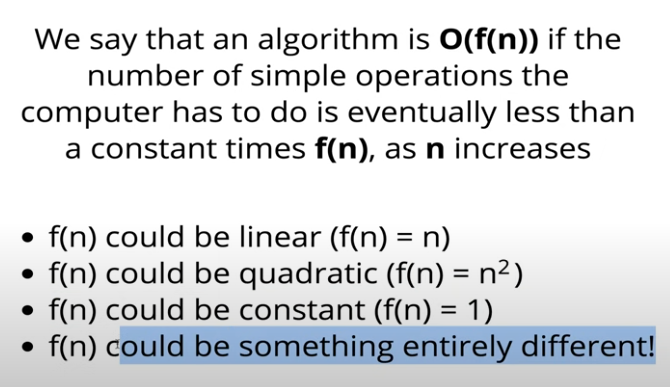
Code Complexity

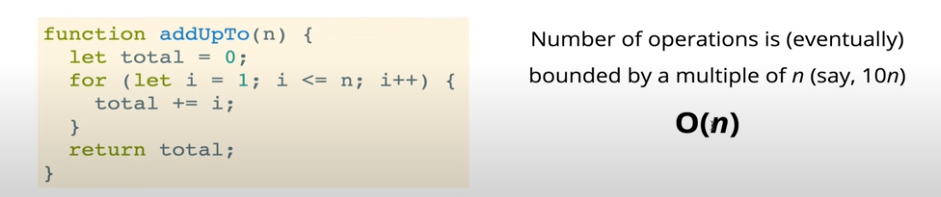
It is basically counting the number of operations.



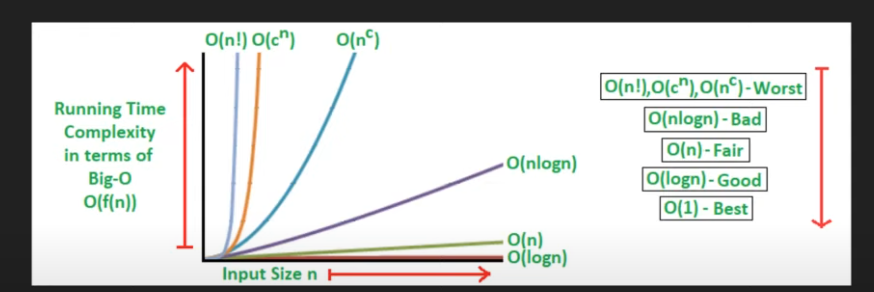
As n grows the amount of time required grows by that function.



Whatever the value of n the number of operations remains same i.e. O(1).

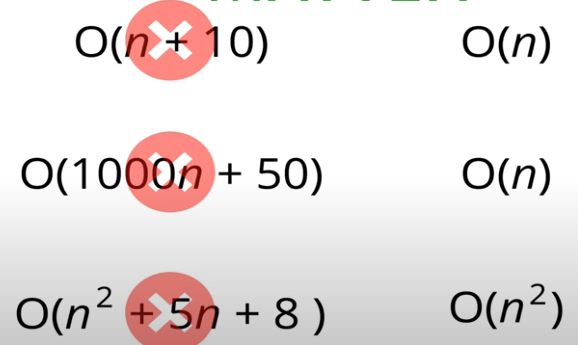


The number of operations increases by n.

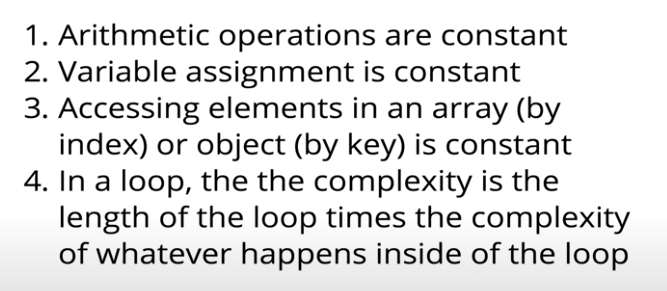


Some famous time complexities.

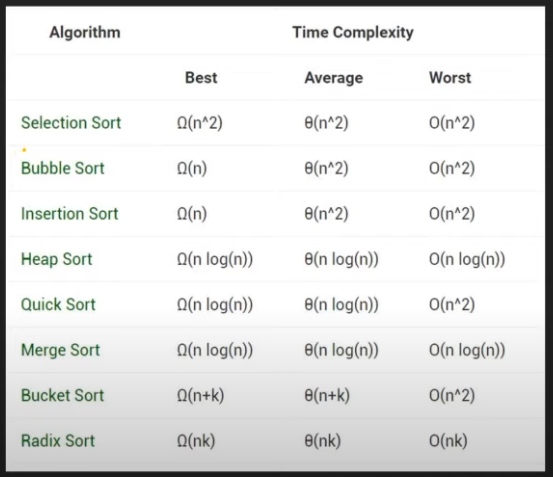
## Rules for Time Complexity

1. Constants don’t matter: O(2n) = O(n), O(13n2) = O(n2), O(220) = O(1),
2. Smaller Terms don’t matter: 

## Big O short hands

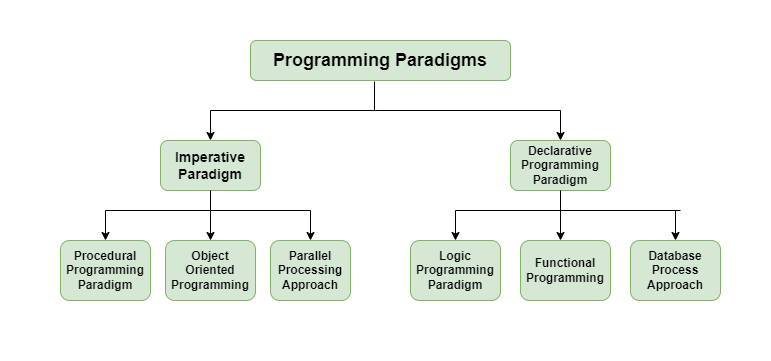


## Time Complexity of various sorting algorithms

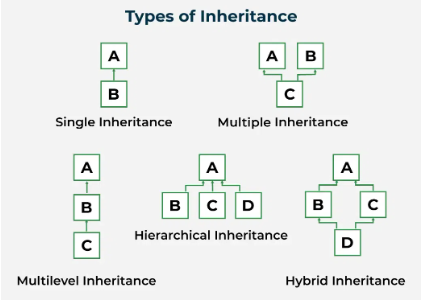


Merge sort is fast but requires more memory, bubble sort is slow but requires low memory.

OOP



## Types of Inheritance



## Limitations of Java

Java does not support multiple inheritance for class neither operator overloading.

## Abstract Class and Method in Java

Abstract class cannot be instantiated on its own it serves as a base class (serves as a modified interface which can have abstract and non-abstract methods, interface only have abstract methods). An **abstract method** is a method that has no implementation in the abstract class, only a method signature (name, parameters, and return type).

## Records in Java

In java records are special classes that auto embeds getters, equals, hashCode, toString methods and constructors.

## Final vs Non-final

**Class:** A final class cannot be inherited by other classes.

**Methods:** A final method cannot be overridden by a subclass.

**Variable:** A final variable becomes constant cannot be reassigned.

## Static vs Non-Static

**Class:** A static class is used by a nested class when it needs to be initialized without its parent being initialized.

**Methods:** A static method belongs to the class itself rather an instance of it, they can be called without creating an instance of class.

**Variable:** A static variable belongs to the class rather than the instance of it and remains same among all instances of it.

## Destructors in Java

Java does not have any destructor but a garbage collection mechanism. You can call super.finalize() to close all memory references.

## Virtual Functions

Virtual functions are the function in child class that override the parent functions.

Pure virtual functions are abstract functions.