

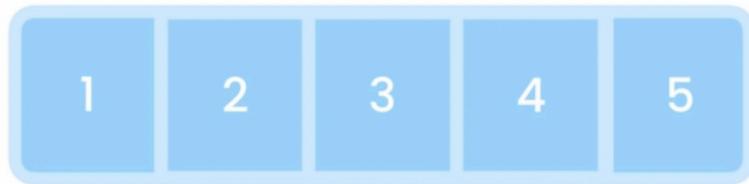
Big O

Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity.

— Wikipedia

We use Big O to describe the performance of an algorithm

ARRAY



array[0]

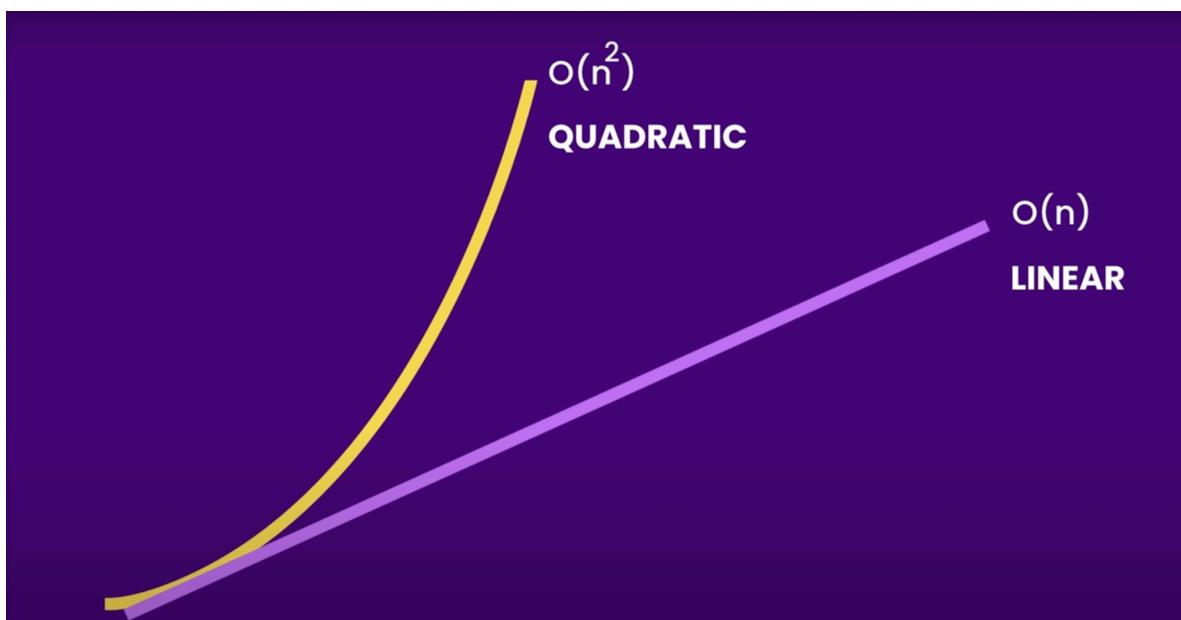
LINKED LIST

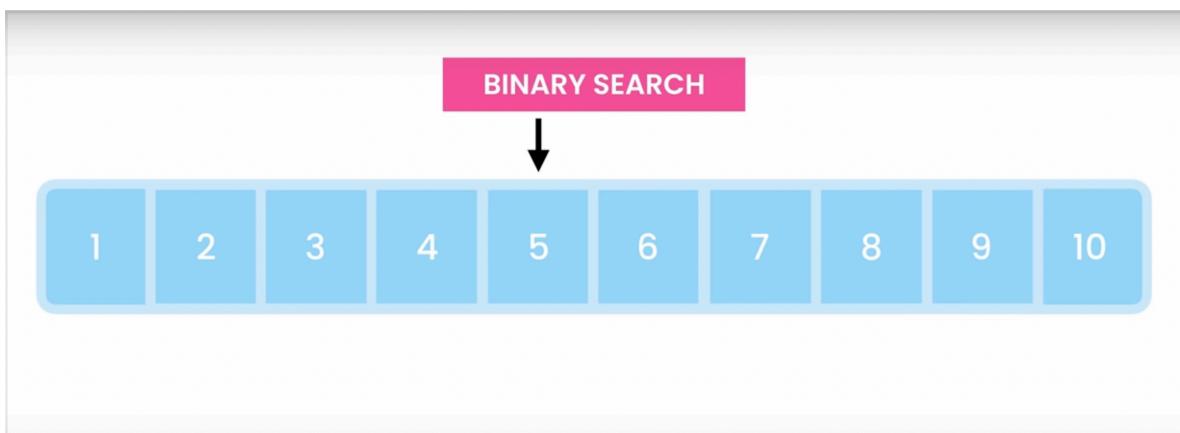
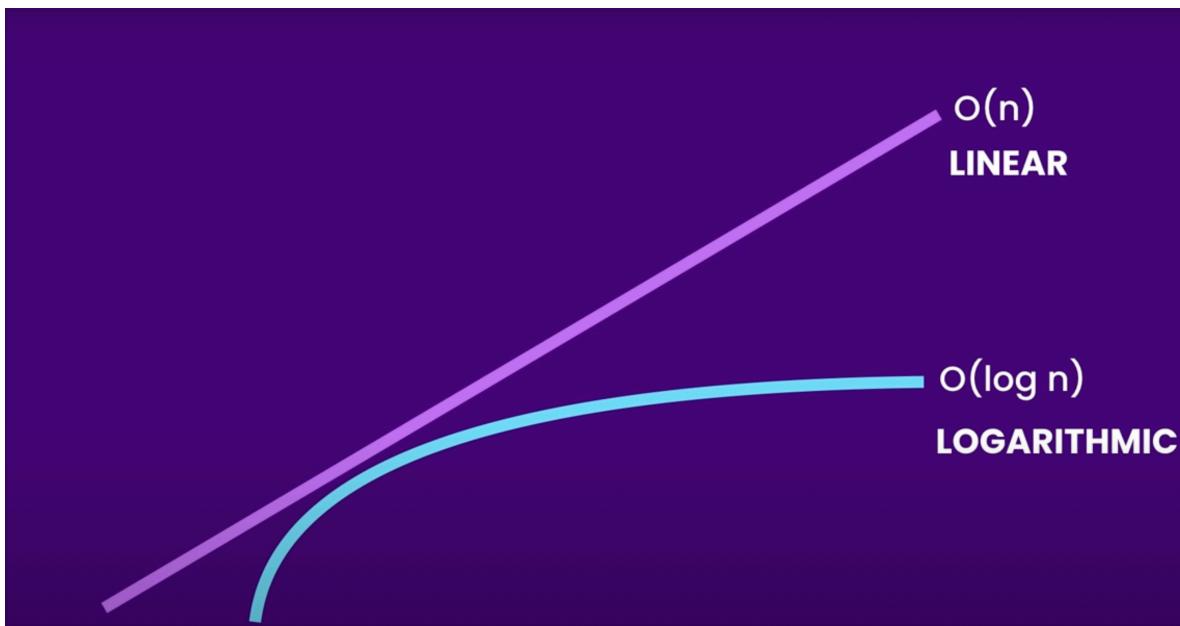


```
public class Main {  
    public void log(int[] numbers) {  
        // O(h)  
        System.out.println(); // O(1)  
        for (int number : numbers) // O(n)  
            System.out.println(number);  
        System.out.println(); // O(1)  
    }  
}
```

```
public class Main {  
  
    public void log(int[] numbers, String[] names)  
        // O(n)  
        for (int number : numbers) // O(n)  
            System.out.println(number);  
  
        for (String name : names) // O(m)  
            System.out.println(name);
```

```
public class Main {  
  
    public void log(int[] numbers) {  
        // O(n ^ 2)  
        for (int first : numbers) // O(n)  
            for (int second : numbers) // O(n)  
                System.out.println(first + ", " + se  
    }  
}
```





CONSTANT

$O(1)$

LOGARITHMIC

$O(\log n)$

LINEAR

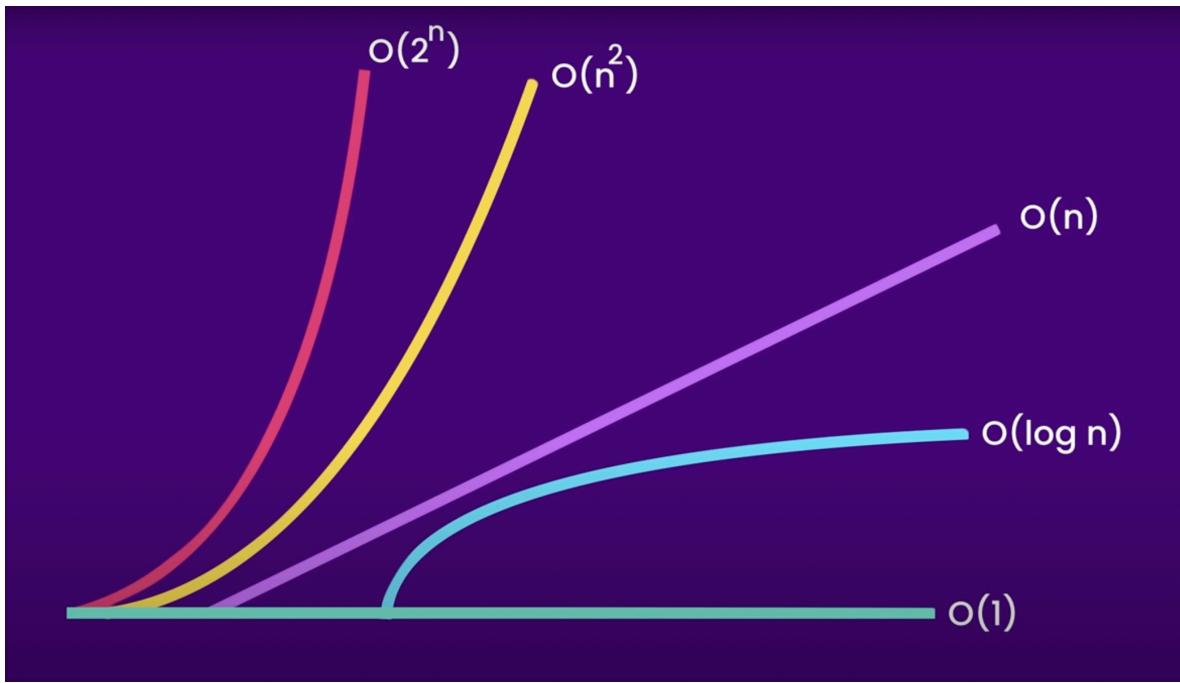
$O(n)$

QUADRATIC

$O(n^2)$

EXPONENTIAL

$O(2^n)$



```
public class Main {
    public void greet(String[] names) {
        // O(n) space
        String[] copy = new String[names.length];
        for (int i = 0; i < names.length; i++)
            System.out.println("Hi " + names[i]);
    }
}
```