# Data Structures + Algorithms

Hector Ferney Garzón Cagua

2020

## Contents

# 1 Array

The javascript **Array** object is a global object that is used in the construction of arrays; which are high-level, list-like objects.

```
const strings = ["a", "b", "c", "d"];
// 4 * 4 = 16 bytes of storage

strings[2];
// expected output: c
```

Every element of an array consumes 4 bytes of memory. If the array has four elements, it's consuming sixteen bytes of memory.

## 1.1 Methods in the array

### 1.1.1 Push

The **push()** methods adds one or more elements to the end of an array and returns the new length of the array. The Big O notation for this method is O(1).

```
const strings = ["a", "b", "c", "d"];

strings.push("e");

console.log(strings);
// expected output: Array ["a", "b", "c", "d", "e"]
```

### 1.1.2 Pop

The **pop()** method removes the **last** element from an array and returns that element. This method changes the length of the array. The Big O notation for this method is O(1).

```
const strings = ["a", "b", "c", "d"];

strings.pop();
// expected output: ["a", "b", "c"]
```

### 1.1.3 Unshift

The **unshift()** methods adds one or more elements to the beginning of an array and returns the new length of the array. The Big O notation for this method is O(n).

```
const strings = ["a", "b", "c", "d"];

strings.unshift("x");
// expected output: ["x", "a", "b", "c", "d"]
```

### 1.1.4 Splice

The **splice()** method changes the contents of an array by removing existing elements and/or adding new elements. The Big O Notation for this method is O(n).

```
const strings = ["a", "b", "c", "d"];

strings.splice(2, 0, "alien")
// expected output: ["a", "b", "c", "alien", "d"]
```

#### Syntax

```
  array.splice(start[, deleteCount[, item1[, item2[, ...s]]]])
```