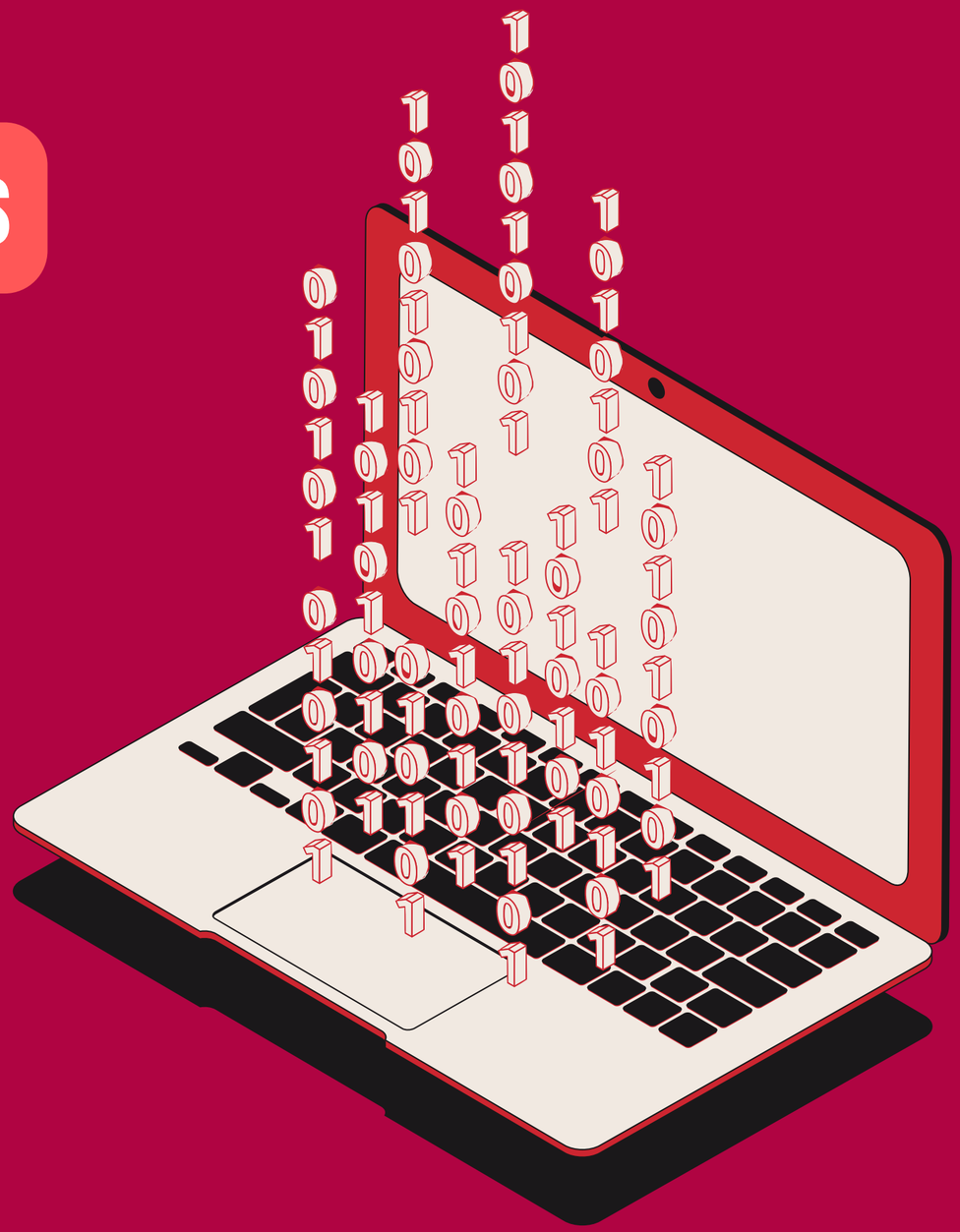
Estructuras de Datos y Algoritmos

Ayudantía 6: Binary Search



Profesor: Yerko Ortiz

Ayudante: Diego Banda

Contacto

diego.banda@mail.udp.cl

darklouds

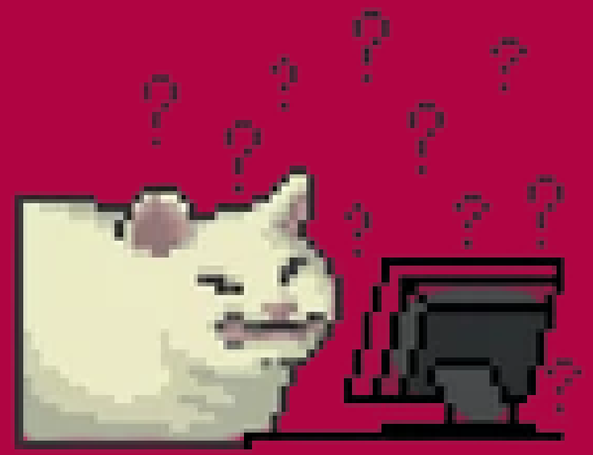
github.com/DiegoBan/EDDA2025-1

¿Qué es binary search?

Como su nombre lo indica, un tipo de búsqueda

2	13	123	456	999

low mid high



¿Por qué nos interesa?

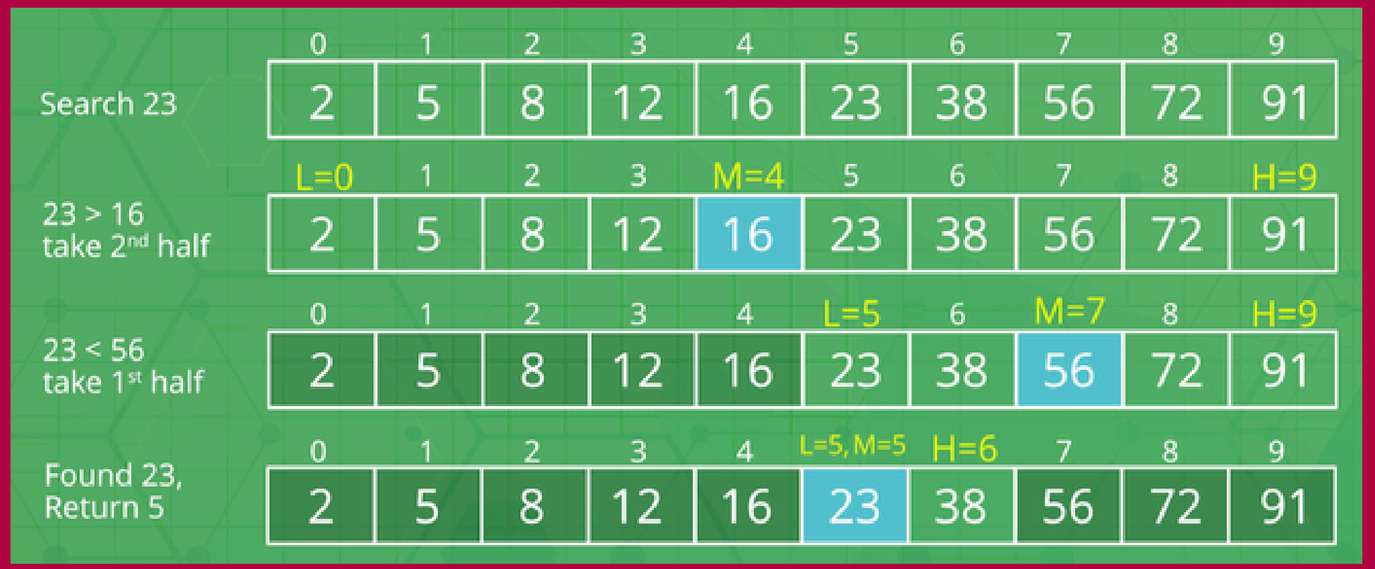
- Muy eficiente: Big O(log(n))
- Divide espacio de búsqueda en cada iteración

Pero... No todo son ventajas...

El arreglo sobre el que se busca DEBE estar ordenado de manera ascendente o descendente.

Funcionamiento

Se compara lo búscado con el número de en medio, lo buscado es menor, se va hacia el arreglo izquierdo, caso contrario, se va hacia el arreglo derecho.



Implementación

Iterativo:

```java
public static int binarySearch(int[] array, int target) {
    int left = 0;
    int right = array.length - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (array[mid] == target) {
            return mid;
        }

        if (array[mid] < target) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    return -1;
}
```

Implementación

Recursivo:

```java
public static int binarySearch(int[] array, int target, int left, int right) {
    if (left > right) {
        return -1;
    }

    int mid = left + (right - left) / 2;

    if (array[mid] == target) {
        return mid;
    } else if (array[mid] > target) {
        return binarySearch(array, target, left, mid - 1);
    } else {
        return binarySearch(array, target, mid + 1, right);
    }
}
```

Ejercicio 1

Ejecuta una búsqueda binaria para los siguientes arreglos:

1. 

Target: 123

2. 

Target: 2

3. 

Target: 12

4. 

Target: 4444

Ejercicio 2

Dado un arreglo con datos duplicados, encontrar el indice de la primera y de la última posición

Input:

- primera linea: N = tamaño arreglo
- N numeros = arreglo

output:

- Indice de la primera y última posición

Ejercicio 3

Dada la siguiente clase:

```java
public class camion{
    String nombre;
    int capacidad;
    camion(String nombre, int capacidad){
        this.nombre = nombre;
        this.capacidad = capacidad;
    }
}
```

Programe una búsqueda binaria según la capacidad del camion, la cual retorne el indice en el cual se encuentra tal camion.
Input: arreglo con camiones, target
Output: indice