

## Trabajo práctico 2 - Recursión, ordenamiento, árboles

### Programación 3 - TUDAI 2021

#### Ejercicio 1.

Implemente un algoritmo recursivo que determine si un arreglo de tamaño  $N$  está ordenado.

1. ¿Qué complejidad  $O$  tiene? (La complejidad en el peor caso)
2. ¿Trae algún problema hacerlo recursivo? ¿Cuál?
3. ¿Qué cambiaría si la estructura fuera una lista en lugar de un arreglo?

#### Ejercicio 2.

Implemente un algoritmo recursivo para buscar un elemento en una lista simple.

Implemente, además, un algoritmo recursivo para buscar un elemento en un arreglo **ordenado ascendentemente** [1 2 3 ...].

1. ¿Qué complejidad  $O$  tienen estos algoritmos?
2. ¿En qué afecta la estructura a estos algoritmos?
3. ¿Cambia algo que la estructura esté ordenada?

#### Ejercicio 3.

Implemente un algoritmo recursivo que convierta un número en notación decimal a su equivalente en notación binaria.

#### Ejercicio 4.

Implemente un algoritmo de ordenamiento por selección en un arreglo.

Implemente un algoritmo de ordenamiento por burbujeo en un arreglo.

1. ¿Qué complejidad  $O$  tienen estos algoritmos?

#### Ejercicio 5.

Implemente un algoritmo de ordenamiento *mergesort* para un arreglo de tamaño  $N$ .

Implemente un algoritmo de ordenamiento *quicksort* para un arreglo de tamaño  $N$ .

1. ¿Cuál es su complejidad en el peor caso?
2. ¿Cuál es su complejidad promedio?

#### Ejercicio 6.

Implemente una función que cree un arreglo de tamaño  $N$  con números aleatorios. Corra los algoritmos de los ejercicios 3 y 4, 10000 veces consecutivas cada uno, y compruebe el tiempo de ejecución. Haga lo mismo con el método `Arrays.sort(...)` de Java, y compare los tiempos obtenidos. Investigue cómo está implementado el `Array.sort()`

#### Ejercicio 7.

Implemente la estructura de Árbol Binario para búsquedas.

Métodos:

- `Integer getRoot()`, `boolean hasElem(Integer)`, `boolean isEmpty()`, `void insert(Integer)`, `boolean delete(Integer)`, `int getHeight()`, `void printPosOrder()`, `void printPreOrder()`, `void printInOrder()`, `List getLongestBranch()`, `List getFrontera()`, `Integer getMaxElem()`, `List getElemAtLevel(int)`
1. ¿Cuál es la complejidad de cada uno de estos métodos?