

ALGORITMIA Y COMPLEJIDAD

LABORATORIO - TEMA 1

EJERCICIO 1

Resuelve las siguientes relaciones de recurrencia:

- a) $T(n) = 2T(n - 1) - T(n - 2) + 8$
- b) $T(n) = 4n + 3 + T(n - 1) + 2T(n - 2)$
- c) $T(n) = 2n + 1 + 2T(n/3)$
- d) $T(n) = T(n/2) + n^2 + 2n$

EJERCICIO 2

Analiza la eficiencia del siguiente código:

```
def inserta_ordenado(lista :list, num :int) -> list:
    """ Devuelve la lista con el número insertado en orden ascendente """
    res = []
    if len(lista) == 0:
        res += [num]
    else:
        pos = 0
        while pos < len(lista) and lista[pos] < num:
            pos += 1
        if pos == len(lista):
            res = lista + [num]
        else:
            res = lista[:pos] + [num] + lista[pos:]
    return res
```

EJERCICIO 3

Analiza la eficiencia del siguiente código que llama a la función del ejercicio 2:

```
pares = []
impares = []
for num in lista:
    if num % 2 == 0:
        pares = inserta_ordenado(pares, num)
    else:
        impares = inserta_ordenado(impares, num)
print(pares)
print(impares)
```

EJERCICIO 4

Analiza la eficiencia del siguiente código recursivo:

```
def buscar(lista :list, num :int) -> bool:
    """ Comprueba si un número está en la lista """
    if len(lista) == 0:
        esta = False
    elif lista[0] == num:
        esta = True
    else:
        esta = buscar(lista[1:], num)
    return esta
```

EJERCICIO 5

Realiza una función que determine si un número recibido como parámetro es primo. Realiza un análisis de eficiencia y complejidad.

EJERCICIO 6

Realiza una función que determine si un número recibido como parámetro es perfecto. Realiza un análisis de eficiencia y complejidad.

EJERCICIO 7

Realiza un programa que pida un número entero positivo al usuario (N) y le diga cuantos primos hay entre 1 y ese número N, y cuantos perfectos hay entre 1 y ese número N. Realiza un análisis de eficiencia y de complejidad.

EJERCICIO 8

Realiza una función recursiva que calcule el número inverso de un número dado. Realiza un análisis de eficiencia y de complejidad.

Ejemplo: 627 → 726

EJERCICIO 9

Realiza una función recursiva que calcule el siguiente sumatorio:

$$S = 1 + 2 + 3 + 4 + \dots + n - 1 + n$$

Realiza un análisis de eficiencia y de complejidad.

Entregables: ejercicio 3, un ejercicio a elegir entre los ejercicios 5, 6 y 7 y un ejercicio a elegir entre los ejercicios 8 y 9.