

Para aprobar es necesario tener al menos 3 ejercicios bien resueltos.

Ejercicio 1

Dado el siguiente método:

```
int simple (int i)
{
    if (i == m)
        return m
    else
        if (i >= m + 1 )
            return 1
        else
            return 2*Simple (i+1) - Simple (i+2);
}
```

- a.- Plantear la función de tiempo de ejecución (no desarrollarla) y explicar brevemente el orden del algoritmo.
b.- Escribir un método que devuelva siempre el mismo resultado que el método anterior, pero tardando **un tiempo de ejecución constante**. Se supone que **m** es una variable de clase (global).

Ejercicio 2

Dada la siguiente Max-heap:

50	30	40	20	5	4	39	6	1	3
1	2	3	4	5	6	7	8	9	10

- a) Indicar el rango de valores que podrían estar en la posición 4 de manera que se siga cumpliendo la propiedad de ser una maxheap.
b) Mostrar los pasos para eliminar el elemento que ocupa la posición 5
c) ¿Puede un vector no vacío ser al mismo tiempo una maxheap y una minheap? Justificar la respuesta.

Ejercicio 3

- a) Implementar un algoritmo para determinar si un grafo no dirigido es un árbol. ¿Cuáles son las condiciones que debe cumplir?
b) ¿Cuál es la complejidad de la solución? Justificar la respuesta.

Ejercicio 4

- a) Dada la siguiente secuencia de números: 7, 4, 6, 15, 20, 9, 3, 12, 14 insertarlos en un árbol AVL (inicialmente vacío), en un orden tal que no sea necesario realizar ninguna rotación.
b) ¿Es siempre posible encontrar ese orden? Justificar la respuesta.