



# **Universitat de Lleida**

## **ESTRUCTURAS DE DATOS**

**GRADO EN TÉCNICAS DE INTERACCIÓN DIGITAL Y  
COMPUTACIÓN**

**Marc Fernández Parra  
German Pérez Bordera**

**Sergio Sayago Barrantes  
Curso 2021-2022**

# Explicación y análisis de algoritmos:

## Caso inicial $O(n^3)$ :

En el código proporcionado, podíamos observar 3 bucles, recorriendo una misma array, el primer bucle era desde  $i = 0$  a `array.length`, el segundo desde la posición actual de la  $i$ , (que se iguala a  $j$ ) hasta el final de la array, y el tercero desde la posición desde la posición actual de la  $i$  hasta la  $k$ .

Este último bucle nos recupera la información de la posición en la que se encuentra la  $k$ , la suma a una variable, y en el segundo bucle comparamos si la suma es mayor o menor que el máximo sumado, en caso de que lo sea lo reemplazamos por lo sumado actualmente, y guardamos la secuencia inicio y fin.

## Caso $O(n^2)$ :

Como podemos intuir leyendo el párrafo anterior, vemos que el tercer bucle sobra, y directamente podemos sumar del valor en la posición " $j$ " con el acumulador, y compararlo con el máximo valor guardado anteriormente, y en caso de ser mayor guardarlo.

## Caso $O(n)$ :

Este caso es algo más complejo, el bucle irá desde  $i=0$  hasta la longitud de la array, y debemos tener en cuenta que pasa cuando la suma actual es negativa. En caso de que sea negativa, la suma actual debe ponerse como 0, y contar el inicio de la posible array de máximo valor el siguiente valor de la  $i$ , guardado en una variable `qt`. Además reiniciaremos un valor auxiliar que nos ayudará a saber si el `qt` es el inicio de la cadena que da el máximo o no.

En el caso de que la suma sea positiva, debemos mirar si es mayor a la anterior máxima. Si lo es, guardaremos el máximo anterior, compararemos el `qt` con el auxiliar, y guardaremos la posición inicial de la cadena. Finalmente, la  $i$  será la posición final siempre que se den estas condiciones.