

# Técnicas de diseño de algoritmos

## Divide y Vencerás

### Ejercicios (Bloque 1): con soluciones

Luis Javier Rodríguez Fuentes  
Amparo Varona Fernández

Departamento de Electricidad y Electrónica  
Facultad de Ciencia y Tecnología, UPV/EHU  
[luisjavier.rodriguez@ehu.es](mailto:luisjavier.rodriguez@ehu.es)  
[amparo.varona@ehu.es](mailto:amparo.varona@ehu.es)

OpenCourseWare 2015  
Campus Virtual UPV/EHU

# Divide y Vencerás – Ejercicios (Bloque 1)

- (B1.1) Demostrar por inducción que el coste amortizado del *quicksort*,  $t_a(n)$ , definido como sigue:

$$t_a(n) = n + \frac{1}{n} \sum_{k=1}^{n-1} (t_a(k) + t_a(n-k))$$

está en  $O(n \log n)$ .

- (B1.2) Escribir en lenguaje Python un algoritmo que obtenga de manera recursiva el valor máximo del tramo  $[izq, der]$  de un vector  $v$ . Expresar el coste temporal del algoritmo en términos de una recurrencia (con caso base y caso general), y resolver ésta por el método que se prefiera.
- (B1.3) Resolver exactamente la siguiente recurrencia mediante el método de la ecuación característica:

$$t(n) = \begin{cases} 1 & n = 1 \\ 4t(n-1) + n^2 & n > 1 \end{cases}$$

- (B1.4) Resolver exactamente la siguiente recurrencia mediante el método de la ecuación característica:

$$t(n) = \begin{cases} 1 & n = 1 \\ 2t(n/2) + \log_2 n & n > 1 \end{cases}$$

# Divide y Vencerás – Ejercicios (Bloque 1)

- (B1.5) Dado un vector  $v$  de longitud  $n$ , aplicando el esquema de diseño *Divide y Vencerás* y sin modificar  $v$ , escribir en lenguaje Python una función que devuelva el valor que ocuparía la posición  $k$  si el vector  $v$  estuviera ordenado. Calcular su complejidad temporal en función de  $n$  y expresarla en notación asintótica. **Sugerencia:** utilícese el ciclo que realiza la partición en el *quicksort*, con  $k \in [izq, der]$ , tal que sólo alguno de los elementos del tramo  $[izq, der]$  puede acabar ocupando la posición  $k$ .
- (B1.6) Escribir en lenguaje Python una función que ordene los elementos del tramo  $[izq, der]$  de un vector  $v$  según el **método de mezcla (mergesort)**, que aplica el siguiente esquema *Divide y Vencerás*: (1) si  $izq < der$ , se consideran las dos mitades del tramo:  $[izq, m]$  y  $[m + 1, der]$ , con  $m = (izq + der)/2$ ; (2) estas dos mitades se ordenan internamente llamando recursivamente a la propia función; y (3) una vez ordenadas, las dos mitades se mezclan en el propio vector  $v$  para producir un único tramo  $[izq, der]$  ordenado. **Sugerencia:** para mezclar las dos mitades, utilícese un vector auxiliar  $w$ , de longitud  $der - izq + 1$ , que finalmente se volcará en el tramo correspondiente del vector  $v$ .