

## Metodología de la Programación y Algoritmia

Convocatoria de Junio 2015

SOLUCIÓN

1.- Distribución de tareas para obtener la mínima penalización

1.a) Descripción del funcionamiento, justificación, pseudocódigo estructura de los nodos.

1.b) Árbol de expansión, tomando como cota la penalización del nodo. Estrategia de extracción: Función de coste y LIFO si coincide el valor de la función. Valores de los nodos, orden de generación, total de nodos generados, podados y expandidos. Interpretar solución obtenida

	Tarea 1	Tarea 2	Tarea 3
Duración (días)	1	2	3
Plazo máximo (días)	2	2	5
Penalización (miles de euros)	5	7	8

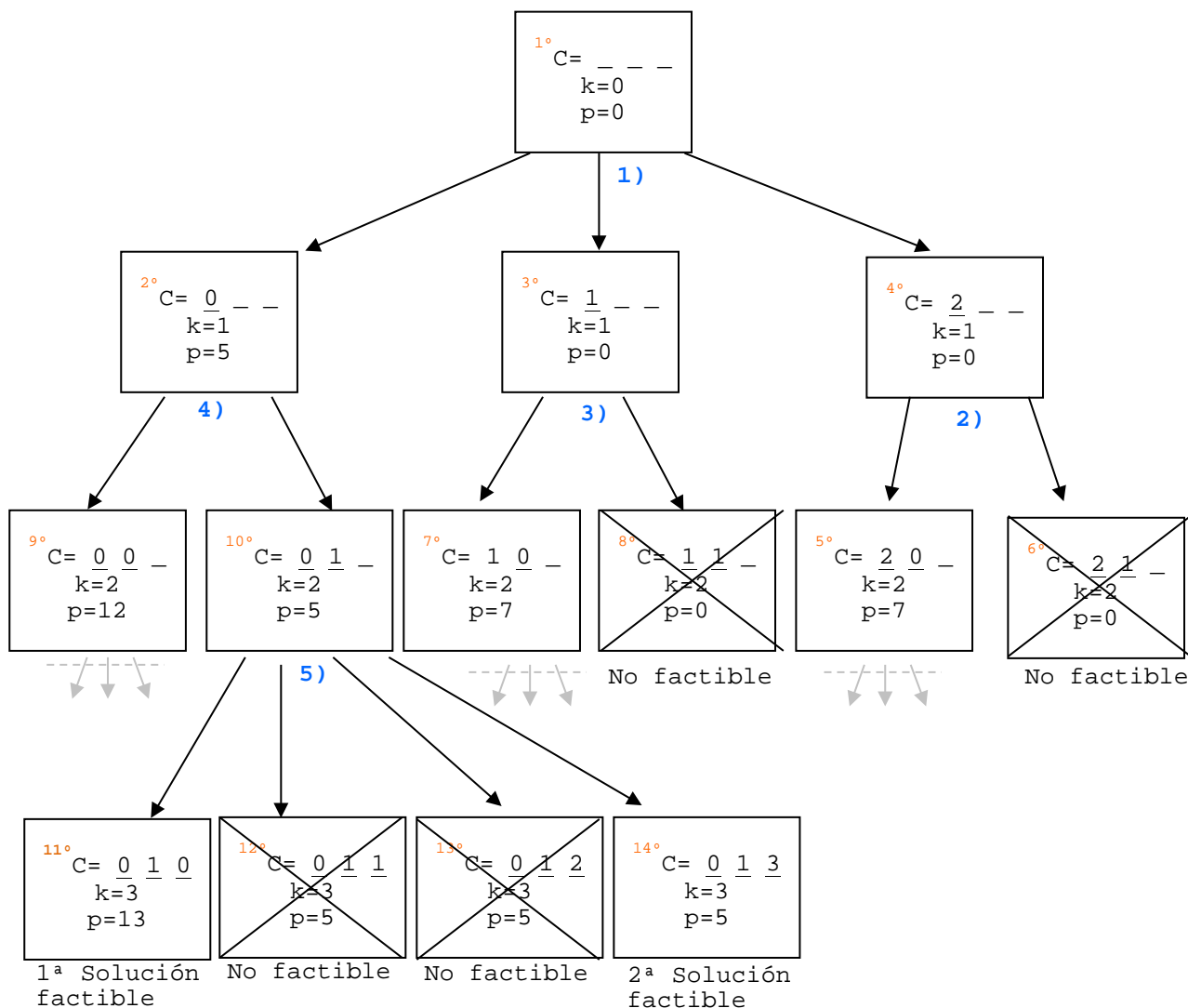
### SOLUCIÓN:

1.a) Este ejercicio se puede resolver utilizando el algoritmo del ejercicio 1 del Tema 7 y que también puede encontrarse en la bibliografía web recomendada (Tema 7).

Guerequeta R., Vallecillo A. (2000), Análisis y diseño de algoritmos. Servicio de publicaciones de la Universidad de Málaga (2000)

1.b) Árbol de expansión.

Según la explicación del algoritmo realizada en el Tema 7. C, indica el día de comienzo de la tarea y p la penalización del nodo, que es la cota de cada nodo, según se indica en el enunciado del problema.



## Metodología de la Programación y Algoritmia

Convocatoria de Junio 2015  
SOLUCIÓN

La evolución de la lista de nodos vivos es la siguiente:

Lista de  
nodos vivos

$\{1^0\}$	$\xrightarrow{1^0}$	$\{\}$
$\{2^0, 3^0, 4^0\}$	$\xrightarrow{4^0}$	$\{2^0, 3^0\}$
$\{2^0, 3^0, 5^0\}$	$\xrightarrow{3^0}$	$\{2^0, 5^0\}$
$\{2^0, 5^0, 7^0\}$	$\xrightarrow{2^0}$	$\{5^0, 7^0\}$
$\{5^0, 7^0, 9^0, 10^0\}$	$\xrightarrow{10^0}$	$\{5^0, 7^0, 9^0\}$
$\{5^0, 7^0, 9^0\}$	$\xrightarrow{7^0}$	$\{5^0, 9^0\}$
$\{5^0, 9^0\}$	$\xrightarrow{5^0}$	$\{9^0\}$
$\{9^0\}$	$\xrightarrow{9^0}$	$\{\}$

La cota del problema la inicializamos a un valor de forma que cuando se encuentre la primera solución factible se actualice. Vamos actualizando la solución del problema conforme encontremos una solución factible mejor que la actual.

cota\_problema =  $+\infty$

cota\_problema = 13 (solución actual: nodo  $9^0$ )

cota\_problema = 5 (solución actual: nodo  $14^0$ )

Número de nodos generados: 14

Número de nodos podados: 3

Número de nodos expandidos: 5

La solución del problema es el nodo  $14^0$ , que indica que la menor penalización que se puede obtener es de 5 y que consiste en realizar la Tarea 2 el día 1, la Tarea 3 el día 3 y no realizar la Tarea 1.

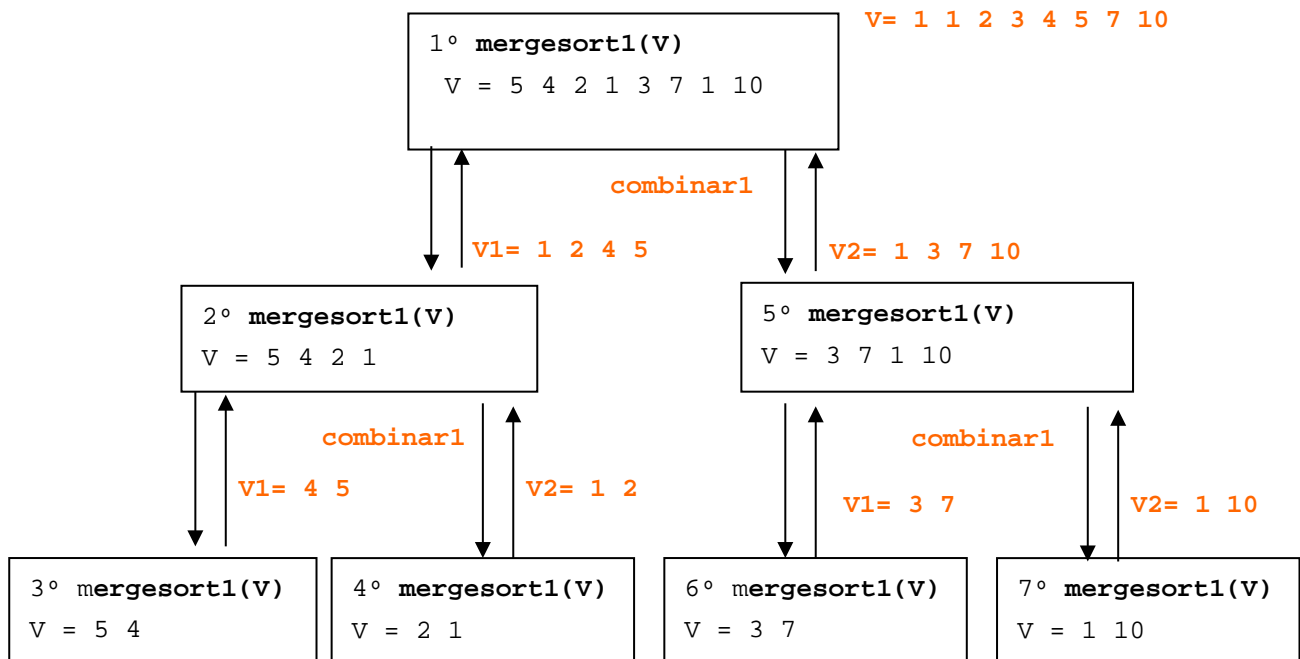
## Metodología de la Programación y Algoritmia

Convocatoria de Junio 2015  
SOLUCIÓN

2.- Algoritmo mergesort: funcionamiento, estrategia, justificación y análisis comparativo.  
Traza para el vector {5, 4, 2, 1, 3, 7, 1, 10}

### SOLUCIÓN:

El algoritmo mergesort se ha visto en teoría dentro del Tema 3 y puede encontrarse en la bibliografía recomendada. La traza utilizando la primera versión del algoritmo es la siguiente:



3.- Dado un algoritmo cuya expresión del tiempo de ejecución es

$$T(n) = \begin{cases} 1 & \text{si } n = 0 \text{ o } n = 1 \\ 4T(n/4) + n & \text{si } n > 1 \end{cases}$$

¿Cuál es su complejidad asintótica? Justifica tu respuesta.

### SOLUCIÓN:

Para calcular la complejidad asintótica se puede hacer aplicando los esquemas o con la técnica del desplegado.

La función T de este ejercicio se ajusta al del tipo de disminución del problema por división, con lo cual podemos aplicar sus esquemas.

$$T(n) = \begin{cases} c_1 & \text{si } 0 \leq n \leq n_1 \\ a \cdot T(n/b) + c \cdot n^k & \text{si } n > n_1 \end{cases} \quad T(n) \in \begin{cases} O(n^k) & \text{si } a < b^k \\ O(n^k \cdot \log n) & \text{si } a = b^k \\ O(n^{\log_b a}) & \text{si } a > b^k \end{cases}$$

Identificamos los elementos:  $c_1 = 1$ ,  $n_1 = 1$ ,  $a = 4$ ,  $b = 4$ ,  $c = 1$ ,  $k = 1$ .

Como  $a = b^k$  ( $4 = 4^1$ ), la complejidad asintótica es  $O(n \log n)$ .