

Metodología de la Programación y Algoritmia

Convocatoria de Junio 2013
SOLUCIÓN

Apellidos _____

Nombre _____

DNI _____

1.- Calcula su complejidad temporal asintótica. Escribe todos los cálculos realizados y justifica la respuesta.

NOTA: El valor del parámetro de la función, x , es una potencia positiva de 2.

SOLUCIÓN:

Como se pide la complejidad asintótica se puede analizar el algoritmo por niveles. La variable que indica el tamaño del problema es x .

	<u>Nivel 1</u>	<u>Nivel 2</u>
función junio2013 (x:entero):entero i, n: entero		
(1) n ← 0	$O(1)$	
(2) i ← 2	$O(1)$	
(3) mientras (i ≤ x) hacer	$O(\log x)$	
(4) i ← 2*i		$O(1)$
(5) n ← n+1		$O(1)$
(6) fmientras		
(7) devolver n	$O(1)$	
ffunción		

Las líneas 4 y 5 siguen la regla 2, correspondiente a la secuencia de instrucciones, con lo cual se coge el máximo que es $\max\{O(1), O(1)\} = O(1)$

	<u>Nivel 1</u>	<u>Nivel 2</u>
función junio2013 (x:entero):entero i, n: entero		
(1) n ← 0	$O(1)$	
(2) i ← 2	$O(1)$	
(3) mientras (i ≤ x) hacer	$O(\log x)$	
(4) i ← 2*i		$O(1)$
(5) n ← n+1		
(6) fmientras		
(7) devolver n	$O(1)$	
ffunción		

Las líneas (3) y (4)-(5) siguen la Regla 4 de los bucles.

La complejidad de la línea 3 depende del n° de iteraciones del bucle, como la variable i se va actualizando multiplicando su valor por una constante el número de iteraciones está en función del logaritmo y es $O(\log x)$. Aplicamos la Regla 4 del producto (correspondiente a la disminución de niveles), con lo cual las líneas (3)-(5) tienen una complejidad $O(\log x) * O(1) = O(\log x)$,

	<u>Nivel 1</u>
función junio2013 (x:entero):entero i, n: entero	
(1) n ← 0	$O(1)$
(2) i ← 2	$O(1)$
(3) mientras (i ≤ x) hacer	$O(\log x)$
(4) i ← 2*i	
(5) n ← n+1	
(6) fmientras	
(7) devolver n	$O(1)$
ffunción	

Finalmente, volvemos a aplicar la regla 2, con lo cual la función junio2013 tiene una complejidad asintótica $O(\log x)$.

Metodología de la Programación y Algoritmia

Convocatoria de Junio 2013
SOLUCIÓN

2.- Dada la función `Obtener(V[n]:entero, izq:entero, der:entero):entero ...`
donde el argumento `V` es un vector ordenado y `izq` y `der` son dos números enteros:

2.a) Determina el tipo de recursión y **justifícalo**.

2.b) Identifica los elementos de esta función con los del esquema general de la recursión determinada en el apartado anterior.

2.c) Realiza la traza de llamadas para $V = \{-2, 0, 1, 4, 7, 8\}$, $izq=1$ y $der=6$, indicando en cada llamada los valores de los parámetros `V`, `izq`, `der`. ¿Cuántas llamadas se realizan a la función? ¿Cuál es el resultado final?

SOLUCIÓN:

2.a) Tipo de recursividad

Recursividad lineal final.

Es lineal porque como mucho en cada una de las ramas del algoritmo se llama la función a sí misma una sola vez y es final porque una vez que llega al caso base se obtiene la solución de la función y ya no quedan operaciones pendientes por realizar.

2.b) Identificación de los elementos con los del esquema general.

Esquema general:

```
función F (x:tipo1):tipo2
  si B(x)
    devolver S(x)
  si no
    devolver F(T(x))
  fsi
ffunción
```

donde

F: nombre de la función recursiva

x: parámetros de entrada

B: condición que determina el caso base

S: solución para el caso base

T: transformación de los parámetros de entrada

Identificación de los elementos:

```
F:      Obtener
x:      V[n], izq, der
B1:     izq<der
S1:     0
B2:     V[centro]=centro
S2:     centro
T1:     V, izq, centro-1
T2:     V, centro+1, der
```

2.c) Traza para $V = \{-2, 0, 1, 4, 7, 8\}$, $izq=1$ y $der=6$,

En todas las llamadas $V=\{-2, 0, 1, 4, 7, 8\}$.

1º) `Obtener(V,1,6)`

centro=3, $V[\text{centro}]=1$

$1 \leq 3$, con lo cual se llama a `Obtener(V,centro+1,der)`

2º) `Obtener(V,4,6)`

centro=5, $V[\text{centro}]=7$

$7 > 5$, con lo cual se llama a `Obtener(V,izq,centro-1)`

3º) `Obtener(V,4,4)`

centro=4, $V[\text{centro}]=4$

$4=4$, con lo cual la función devuelve un 4.

Nº llamadas a la función: 3

Resultado final: 4

Metodología de la Programación y Algoritmia

Convocatoria de Junio 2013

SOLUCIÓN

3.- Describe el funcionamiento del algoritmo de ordenación rápida (quicksort), indica a qué técnica de diseño de algoritmos se ajusta y realiza un análisis completo sobre su complejidad asintótica.

Dibuja el árbol de llamadas del algoritmo para el vector {5 8 10 6 7 4 2} tomando como pivote la mediana de los elementos que ocupan las posiciones primera, última y centro del vector. Indica el orden en que se producen las llamadas, los valores del vector y de los pivotes.

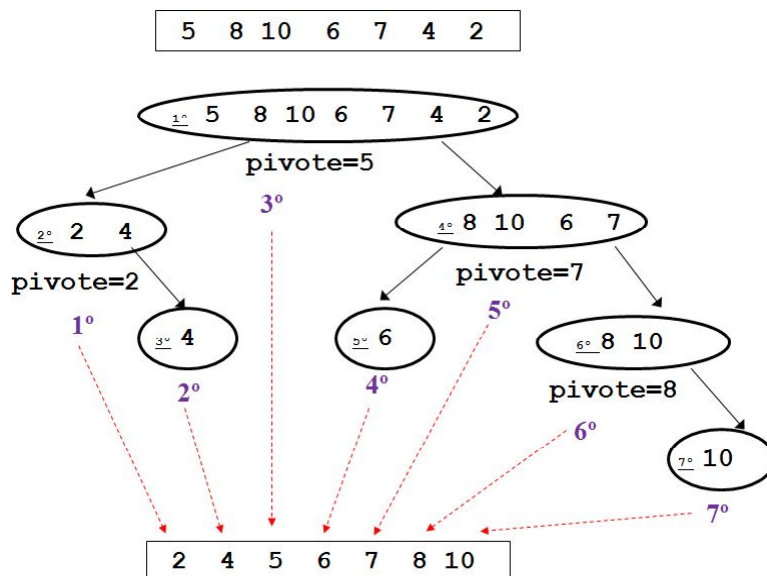
(2.25 puntos)

SOLUCIÓN:

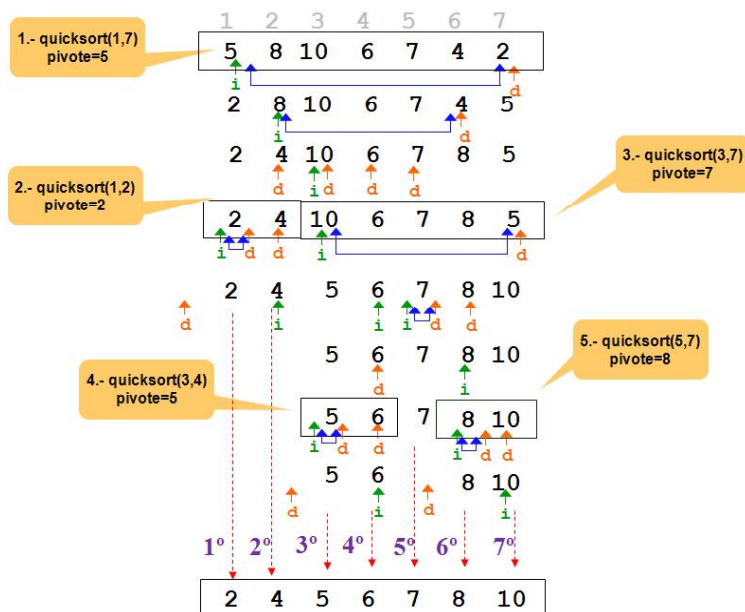
Describir el funcionamiento, técnica de diseño y realizar análisis completo de la complejidad asintótica

Árbol de llamadas

Opción 1:



Opción 2: También se podría haber hecho haciendo la secuencia de llamadas según el pseudocódigo.



Metodología de la Programación y Algoritmia

Convocatoria de Junio 2013

SOLUCIÓN

4.- Compañía aérea

4.a) Describe **detalladamente** el funcionamiento de un algoritmo **lo más eficiente** posible para obtener los precios más baratos entre todo par de ciudades.

4.b) Indica qué tipo de estrategia de programación utiliza el algoritmo propuesto y por qué.

4.c) Aplica el algoritmo propuesto a los precios dados para la semana actual, indica todos los resultados parciales que se van obteniendo y el resultado final.

SOLUCIÓN:

4.a y 4.b).

Se utiliza el algoritmo de Floyd, visto en el tema de Programación Dinámica. Hay que explicar detalladamente el funcionamiento del algoritmo para que a partir de la explicación se pueda saber cómo aplicas el algoritmo.

4.c)

Matriz de adyacencia

0	90	30	150	Inf
120	0	70	125	Inf
20	Inf	0	90	50
25	80	70	0	30
Inf	30	Inf	30	0

k=1

0	90	30	150	Inf
120	0	70	125	Inf
20	110	0	90	50
25	80	55	0	30
Inf	30	Inf	30	0

k=2

0	90	30	150	Inf
120	0	70	125	Inf
20	110	0	90	50
25	80	55	0	30
150	30	100	30	0

k=3

0	90	30	120	80
90	0	70	125	120
20	110	0	90	50
25	80	55	0	30
120	30	100	30	0

k=4

0	90	30	120	80
90	0	70	125	120
20	110	0	90	50
25	80	55	0	30
55	30	85	30	0

k=5

0	90	30	110	80
90	0	70	125	120
20	80	0	80	50
25	60	55	0	30
55	30	85	30	0

Así, los precios de los vuelos más baratos entre todo par de ciudades es:

	Londres	París	Madrid	Roma	Praga
Londres	0	90	30	110	80
París	90	0	70	125	120
Madrid	20	80	0	80	50
Roma	25	60	55	0	30
Praga	55	30	85	30	0