

# (UT3) UT2 PD3

## Ejercicio 1

### Weiss, Ejercicio 5.4

Suponga que  $T_1(N)$  y  $T_2(N) = O(F(N))$ . ¿Cuáles de las siguientes afirmaciones son ciertas?

- a.  $T_1(N) + T_2(N) = O(F(N))$  - Verdadero porque  $x^2 + x^2 = 2x^2$
- b.  $T_1(N) - T_2(N) = O(F(N))$  - ¿Falso?
- c.  $T_1(N)/T_2(N) = O(1)$  - Verdadero porque  $x^2/x^2 = 1$
- d.  $T_1(N) = O(T_2(N))$  - ¿Falso?

### Weiss, Ejercicio 5.5

Resolver un problema requiere de ejecutar un algoritmo  $O(N)$  y después otro algoritmo  $O(N)$ . ¿Cuál es el coste total de resolver el problema? El coste total de resolver el problema es de  $O(N)$  porque para saber el coste total solo debemos sumar los órdenes de los algoritmos que componen la solución, por lo que  $2N$  sigue siendo de orden  $N$  una vez lo simplificamos.

### Weiss, Ejercicio 5.6

Agrupe las siguientes funciones según su equivalencia desde el punto de vista de análisis de Big O.

$xyx^2 - x$ ,  $O(N)$ ,  $x^2$ ,  $O(N^2)$ ,  $x^4$ ,  $O(N^4)$ ,  $X^4/(x - 1)$ ,  $O(\log(N))$ .

### Weiss, Ejercicio 5.10

Complete la figura 5.10 con estimaciones para los tiempos de ejecución que eran demasiado largos como para simularlos. Interpole los tiempos de ejecución para los cuatro algoritmos y estime el tiempo requerido para calcular la suma máxima de subsecuencia contigua de 10.000.000 de números. ¿Qué suposiciones ha hecho?

N	$O(N^3)$	$O(N^2)$	$O(N \log N)$	$O(N)$
10	0,000001	0,000000	0,000001	0,000000
100	0,000288	0,00019	0,000014	0,000005
1.000	0,223111	0,001630	0,000154	0,000053

N	$O(N^3)$	$O(N^2)$	$O(N \log N)$	$O(N)$
10.000	218	0,133064	0,001630	0,000533
100.000	$218 * 1.000 = 218.000$	13,17	0,017467	0,005571
1.000.000	218,000,000	$13.17 * 100 = 131,7$	0,185363	0,056338
10.000.000	218,000,000,000	$131,7 * 100 = 13.170$	1,946311	0.56338

En general, cuando la entrada se multiplica por un factor  $f$ , el tiempo de ejecución se multiplica por un factor  $f^z$ . Sea  $z$  un número entero distinto de 0.

### Weiss, Ejercicio 5.11

Evalúe directamente el sumatorio triple que precede al Teorema 5.1. Verifique que las respuestas son idénticas.

### Weiss, Ejercicio 5.12

Un algoritmo requiere 0,4 ms para un tamaño de la entrada de 100. ¿Cuánto tiempo requerirá para un tamaño de entrada igual a 500 (suponiendo que los términos de menor orden sean despreciables), si el tiempo de ejecución es:

- a. lineal: 2 s
- b.  $O(N \log N)$ :
- c. cuadrático: 10 s
- d. cúbico: 50 s

### Weiss, Ejercicio 5.13

Para los algoritmos típicos que emplee para realizar cálculos a mano, determine el tiempo de ejecución necesario para

- a. Sumar dos enteros de  $N$  dígitos:  $O(N)$

```

SumarDosEnteros(x, y)
COM
    sumarUnidades
    sumarDecenas
    sumarCentenas

```

```
... // depende de N la cantidad de sumas → O(N)
FIN
```

b. Multiplicar dos enteros de N dígitos: O(N)

```
MultiplicarDosEnteros(x, y)
COM
  Para cada dígito n en x.reverso:
    multiplicar con y.unidad
  Para cada dígito n en x.reverso:
    multiplicar con y.decena
  ... // depende de N la cantidad de veces que multiplicamos → O(N)

  sumar resultados de las multiplicaciones // O(1)
FIN
```

### Weiss, Ejercicio 5.14

```
public static int maxSumaMaxima(int[] arr) {

    // Inicializar la suma máxima y la suma actual con el primer elemento del Ar
    int maxSum = arr[0];
    int currentSum = 0;

    // Recorrer el Array elemento por elemento
    for (int i = 0; i < arr.length; i++) {

        // Actualizar la suma actual sumándole el elemento actual
        currentSum += arr[i];

        // Si la suma actual es mayor que la suma máxima, actualizar la suma má
        maxSum = Math.max(maxSum, currentSum);

        // Si la suma actual es negativa, reiniciarla a 0
        if (currentSum < 0) {
            currentSum = 0;
        }
    }
}
```

```
// Devolver la suma máxima encontrada
return maxSum;
}
```

### Weiss, Ejercicio 5.15

Para 1.000 elementos, nuestro algoritmo tarda 10 segundos en ejecutarse en la máquina A, pero ahora sustituimos la máquina A por la máquina B que es 3 veces más rápida.

¿Aproximadamente cuánto tiempo tardará el algoritmo en ejecutarse en la máquina B para 2.500 algoritmos si el algoritmo es:

- a. lineal:  $10 * 2,5 = 25$  segundos en la máquina A.  $25 / 3 = 8,333$  segundos en máquina B.
- b. cuadrático:  $10 * (2,5)^2 = 62,5$  segundos en A.  $62,5 / 3 = 20,833$  segundos en B.
- c.  $O(N^3)$ :  $10 * (2,5)^3 = 156,25$  segundos en A.  $156,25 / 3 = 52,083$  segundos en B.
- d.  $O(N \log N)$ :

### Weiss, Ejercicio 5.16

Un algoritmo requiere un tiempo de ejecución de 0,5 ms para un tamaño de entrada igual a 100. ¿Qué tamaño de problema puede resolver en un minuto (suponiendo que los términos de orden inferior sean despreciables) si el tiempo de ejecución es:

- a. lineal: 100 en 0,5 ms. 200 en 1 s. 12.000 en 1 minuto.
- b.  $O(N \log N)$ :
- c. cuadrático: ...
- d. cúbico: ...