

# El libro de FAA

Jesús García Gutiérrez



# Capítulo 1

## Operación elemental

Una operación elemental es una operación que tarda  $t$  unidades de tiempo. Siempre tarda  $t$ , sin importar cuándo la hagamos.

Si consideramos la suma como una operación elemental, el ordenador tardará  $t$  unidades de tiempo en hacer una. Hacer dos sumas le tomaría  $2 \cdot t$  unidades de tiempo. Hacer  $n$  sumas le tomaría  $n \cdot t$  unidades de tiempo.

Así, tenemos una manera sencilla de calcular el tiempo que tardará el ordenador en hacer  $n$  sumas. Realmente, el ordenador no tarda un tiempo exacto  $t$  en hacer una suma (puede variar por circunstancias), pero esta simplificación nos hace posible analizar los algoritmos sin preocuparnos de detalles de software o de hardware.

Si también definimos la multiplicación como una operación elemental, esto significa que el ordenador también tardará  $t$  unidades de tiempo en hacer una multiplicación. Podemos decir entonces lo siguiente.

$$t_{\text{suma}} = t_{\text{multiplicacion}} = t \quad (1.1)$$

No solo consideraremos la suma y la multiplicación como operaciones elementales. Consideraremos como operaciones elementales todas las operaciones de la siguiente tabla.

Suma
Resta
Multiplicación
División
Módulo
Asignación
Llamada a una función
Acceso a un elemento del array

El ordenador tarda  $t$  unidades de tiempo en realizar cualquiera de estas operaciones.

## Capítulo 2

# Relación de recurrencia

### 2.1. Ejercicio del 02/04/2025

Hallar la eficiencia de un algoritmo dado. Supongamos que ya nos dan la función a trozos del  $T(n)$ .

$$T(n) = \begin{cases} 1 & n \leq 1 \\ T(\frac{n}{2}) + 2 & n > 1 \end{cases}$$

Primero aislamos la relación de recurrencia. Las relaciones de recurrencia tienen en la parte derecha de la igualdad una o más  $T$ , a diferencia de los casos base, que no tienen ninguna  $T$  en dicha parte.

$$T(n) = T(\frac{n}{2}) + 2 \quad (2.1)$$

En este caso, podemos aplicar tanto el método de expansión de recurrencia como el teorema maestro. Esto es así porque solo hay una  $T$  en la parte derecha de la igualdad. Si hubiera más de una  $T$  en dicha parte, no podríamos aplicar ninguno de estos métodos.

#### 2.1.1. Método de expansión de recurrencia

El método de expansión de recurrencia consiste en hallar el tiempo para problemas cada vez más pequeños, y calcular el  $T(n)$  utilizando cada uno de estos hallazgos.

Si nos fijamos en la ecuación 1.1, vemos  $T(\frac{n}{2})$ . ¡Vamos a hallarlo usando  $T(n)$ !

$$\begin{aligned} T(\frac{n}{2}) &= T(\frac{\frac{n}{2}}{2}) + 2 \\ T(\frac{n}{2}) &= T(\frac{n}{4}) + 2 \end{aligned}$$

Listo, hemos desarrollado  $T(\frac{n}{2})$ . Ahora podemos sustituir la expresión que hemos hallado en la ecuación 1.1.

$$\begin{aligned} T(n) &= (T(\frac{n}{4}) + 2) + 2 \\ T(n) &= T(\frac{n}{4}) + 4 \end{aligned} \tag{2.2}$$

Perfecto, la ecuación 1.2 es una nueva fórmula para el  $T(n)$ .

Ahora solo tenemos que seguir el mismo procedimiento que antes, hasta que veamos un patrón claro y podamos escribirlo en una ecuación. ¡Vamos con el  $T(\frac{n}{4})$  que aparece en la ecuación 1.2!

$$\begin{aligned} T(\frac{n}{4}) &= T(\frac{\frac{n}{4}}{2}) + 2 \\ T(\frac{n}{4}) &= T(\frac{n}{8}) + 2 \end{aligned}$$

Listo, sustituimos en la ecuación 1.2.

$$\begin{aligned} T(n) &= (T(\frac{n}{8}) + 2) + 4 \\ T(n) &= T(\frac{n}{8}) + 6 \end{aligned} \tag{2.3}$$

Ya lo vas pillando, ¡a por el  $T(\frac{n}{8})$ !

$$\begin{aligned} T(\frac{n}{8}) &= T(\frac{\frac{n}{8}}{2}) + 2 \\ T(\frac{n}{8}) &= T(\frac{n}{16}) + 2 \end{aligned}$$

Sustituimos  $T(\frac{n}{8})$  en la ecuación 1.3.

$$\begin{aligned} T(n) &= (T(\frac{n}{16}) + 2) + 6 \\ T(n) &= T(\frac{n}{16}) + 8 \end{aligned} \tag{2.4}$$

Ya hemos hallado el  $T(n)$  utilizando varios subproblemas. Si nos fijamos en las ecuaciones 1.1, 1.2, 1.3 y 1.4, ¿veis el patrón? Vamos a escribirlo utilizando una variable llamada  $i$ .

$$T(n) = T(\frac{n}{2^i}) + 2 \cdot i \tag{2.5}$$

Estupendo, la ecuación 1.5 es muy bonita y parece que ya somos informáticos teóricos cum laude, pero ocurre una desgracia: no está solo en función de  $n$ , también está en función de  $i$ . Tenemos que transformar  $i$  en  $n$  de algún modo.

En la ecuación 1.5 fijémonos en la expresión  $T(\frac{n}{2^i})$ . Solo tenemos que igualar el paréntesis al valor de  $n$  que sea caso base. Si volvemos a la función a trozos, vemos que el caso base tiene  $n = 1$ , y por tanto podemos escribir lo siguiente.

$$\frac{n}{2^i} = 1$$

Podríamos pensar que  $n = 0$  es también un caso base, pero no trabajaremos nunca con problemas de tamaño igual a cero. Por eso ponemos 1 y no 0 en la igualdad. Seguimos desarrollando y aplicando la idea de logaritmo.

$$n = 2^i$$

$$\log_2(n) = i$$

$$i = \log_2(n)$$

Listo, tenemos  $i$  en función de  $n$ . Sustituimos la  $i$  en la ecuación 1.5.

$$T(n) = T\left(\frac{n}{2^{\log_2(n)}}\right) + 2 \cdot \log_2(n)$$

El denominador se puede seguir simplificando matemáticamente.

$$2^{\log_2(n)} = n^{\log_2(2)}$$

$$2^{\log_2(n)} = n^1$$

Maravilloso, ahora el denominador es simplemente  $n$ . Volvemos a donde estábamos antes.

$$T(n) = T\left(\frac{n}{n}\right) + 2 \cdot \log_2(n)$$

$$T(n) = T(1) + 2 \cdot \log_2(n)$$

Sustituimos  $T(1)$ , sabiendo por la función a trozos que  $T(1) = 1$

$$T(n) = 1 + 2 \cdot \log_2(n)$$

$$T(n) = 2 \cdot \log_2(n) + 1 \tag{2.6}$$

Listo, qué bonita nos ha quedado la fórmula 1.6. Esta nos da el número de operaciones elementales dada una  $n$ , y no tenemos que hacer tiempos de otros subproblemas como ocurría con la relación de recurrencia originalmente.

También podemos hablar ya del orden de complejidad. Podemos ver que...

$$T(n) \in O(\log_2(n))$$