Jaime Hernández Delgado
48776554 W

Ejercicio 2 → abstracto   Tamaño → n (cte.) depende únicamente de n

Para esta función podemos ver que tiene un bucle for que imprime "*" y otro bucle for que llama a abstracto dividiendo el parámetro entre 2 cuatro veces, tampoco podemos ver que haya ni peor ni mejor caso

Para el primer bucle tenemos que

$$\sum_{i=1}^{n-1} 1 = \frac{(n+\cancel{1}-\cancel{1})(1+1)}{2} = \frac{2n}{2} = n \in O(n)$$

↳ como es una serie aritmética usamos

$$\sum_{i=1}^{n} i = \frac{(n+1)(a_0+a_n)}{2}$$

Para el segundo bucle sin embargo vemos que se ejecuta cuatro veces teniendo así como ecuación de recurrencia :

$$T(n) \begin{cases} 1 & n \leq 1 \\ n + 4T\left(\frac{n}{2}\right) & n > 1 \end{cases}$$

(1) $T(n) = n + 4T\left(\frac{n}{2}\right)$

(2) $= n + 4\left(\frac{n}{2} + 4T\left(\frac{n}{4}\right)\right) = 3n + 4^2 T\left(\frac{n}{2^2}\right)$

(3) $= 3n + 4^2\left(\frac{n}{2^3} + 4T\left(\frac{n}{2^3}\right)\right) = 5n + 4^3 T\left(\frac{n}{2^3}\right)$

(k) $= (2k-1)n + 4^k T\left(\frac{n}{2^k}\right)$

igualamos a 1 para obtener el caso base → $\frac{n}{2^k} = 1$; $n = 2^k$; $k = \log_2(n)$

$(2(\log_2(n))-1)n + 4^{\log_2(n)} T(1)$
① ②

① $n\log_2(n)$

② $2^{2\log_2(n)}$; $a^{\log_a b} = b$; n^2

Aplicamos la propiedad de los logaritmos

Tenemos entonces $O(n\log_2(n))$ y $O(n^2)$

linealogarítmica

polinómico

como $O(n\log_2(n)) \subset O(n^2)$ la complejidad de esta función es $O(n\log_2(n))$ //

La función lineal-algorítmica crece más rápido que la polinómica