

Ejercicio 4 (*ComplexityQuest*) ★

Calcule la complejidad de un algoritmo que utiliza $T(n)$ pasos para una entrada de tamaño n , donde T cumple:

- | | | |
|-------------------------------|-------------------------------|-------------------------------|
| 1) $T(n) = T(n-2) + 5$ | 5) $T(n) = 2T(n-1)$ | 9) $T(n) = 2T(n-4)$ |
| 2) $T(n) = T(n-1) + n$ | 6) $T(n) = T(n/2) + n$ | 10) $T(n) = 2T(n/2) + \log n$ |
| 3) $T(n) = T(n-1) + \sqrt{n}$ | 7) $T(n) = T(n/2) + \sqrt{n}$ | 11) $T(n) = 3T(n/4)$ |
| 4) $T(n) = T(n-1) + n^2$ | 8) $T(n) = T(n/2) + n^2$ | 12) $T(n) = 3T(n/4) + n$ |

Intentar estimar la complejidad para cada ítem directamente y luego calcularla utilizando el teorema maestro de ser posible. Para simplificar los cálculos se puede asumir que n es potencia o múltiplo de 2 o de 4 según sea conveniente.

Todos los demás no se pueden calcular con el teorema maestro.

6

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

$$\underbrace{\log_2 1}_{\log_b a} = 0 < \underbrace{1}_{\check{c}} \implies T(n) = \Theta(n)$$

7

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

$$\underbrace{\log_2 1}_{\log_b a} = 0 < \underbrace{0.5}_{\check{c}} \implies T(n) = \Theta(\sqrt{n})$$

8

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

$$\underbrace{\log_2 1}_{\log_b a} = 0 < \underbrace{2}_{\check{c}} \implies T(n) = \Theta(n^2)$$

10

$$T(n) = 2T\left(\frac{n}{2}\right) + \log n, \text{ acotable por } T(n) = 2T\left(\frac{n}{2}\right) + O(\sqrt{n})$$

$$\underbrace{\log_2 2}_{\log_b a} = 1 > \underbrace{0.5}_{\check{c}} \implies T(n) = \Theta(n)$$

12

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

$$\underbrace{\log_4 3}_{\log_b a} < \underbrace{1}_{\check{c}} \implies T(n) = \Theta(n)$$