

Questão 09

* Algoritmo A:

$$T(n) = 5T\left(\frac{n}{2}\right) + O(n)$$

* Algoritmo B:

$$T(n) = 2T(n-1) + O(1)$$

* Algoritmo C:

$$T(n) = 9T\left(\frac{n}{3}\right) + O(n^2)$$

Complexidade A:

$$a=5 \quad b=2$$

1º caso: $\Rightarrow \log_b a = \log_2 5 = 2,32...$

~~...~~

$$f(n) = O(n)$$

Aplicando:

~~...~~

$$O(n) = O(n^{2,32... - \epsilon})$$

$$\epsilon = \log_2 5 - 1$$

$$\rightarrow T(n) = O(n^{\log_b a})$$

$$\approx O(n^2)$$

Algoritmo C:

$$a = 9 \quad b = 3$$

$$f(n) = O(n^2)$$

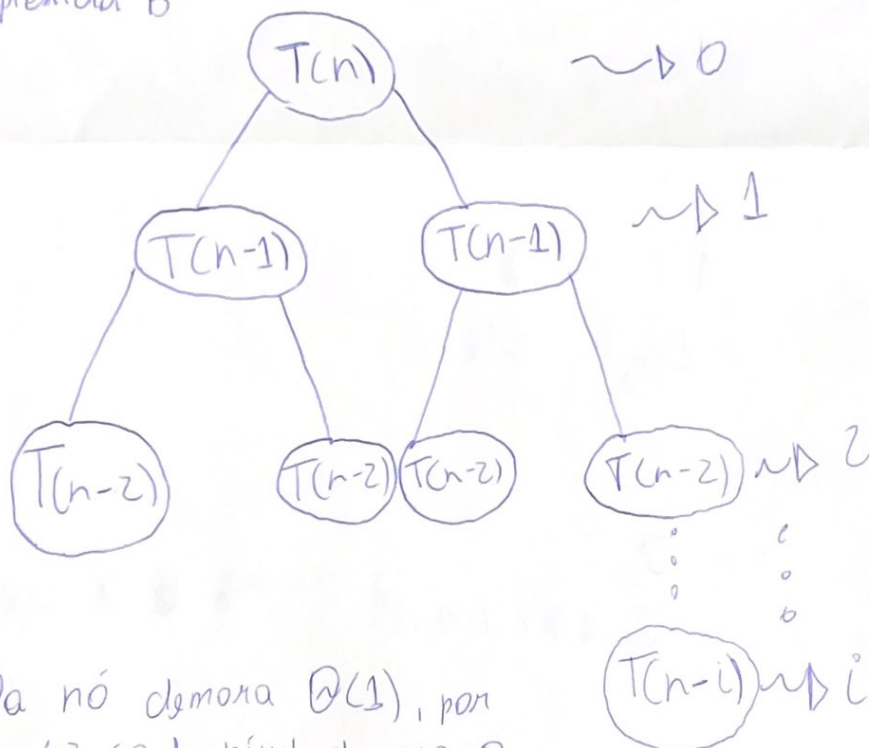
$$\log_b a = \log_3 9 = 3^x = 9 \Rightarrow 3^x = 3^2 \Rightarrow \boxed{x=2}$$

2º caso:

$$f(n) = O(n^{\log_b a}) \rightarrow T(n) = O(n^{\log_b a} \log n)$$

$$O(n^2) = O(n^2) \rightarrow T(n) = O(n^2 \log n)$$

Complexidade B:



Cada nó demora $O(1)$, por
sua vez cada nível demora o
nº de nós $\times O(1)$

$$\boxed{2^i} \cdot O(1) \downarrow \rightarrow 2^{n-1}$$

$$\begin{aligned} \text{P.G.} &: O(1) + 2 \cdot O(1) + \dots + 2^{n-1} \cdot O(1) = \\ &= 4 \cdot O(1) \dots 2^{n-1} \cdot O(1) = \\ &= \frac{2^n - 1}{2 - 1} = \boxed{\Theta(2^n)} \end{aligned}$$

$$\begin{aligned} T(n-i) &= 1 \rightarrow \text{Último nó} \\ n-i &= 1 \\ i &= n-1 \\ \boxed{n = i+1} \end{aligned}$$

Algoritmos A, B e C:

$\Theta(n^{\log_2 5})$, $\Theta(2^n)$ e $\Theta(n^2 \log n)$

$\Theta(1)$

$\Theta(\log n)$

~~$\Theta(n)$~~

$\Theta(n)$

$\Theta(n \log n)$

$\Theta(n^2)$

$\Theta(n^3)$

\vdots

$\Theta(2^n)$

+ rápido

- rápido

Algoritmo A seria
o melhor pra escolher
pois é o mais rápido
dos 3