AULA 5 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS RECURSIVOS

***** Entregue, num ficheiro ZIP, este guião preenchido e o código desenvolvido *****

Implemente os seguintes **algoritmos recursivos** – **sem recorrer a funções de arredondamento** (floor e ceil) – e analise o **número de chamadas recursivas** executadas por cada algoritmo.

$$\mathrm{T}_1(\mathrm{n}) = \begin{cases} 0, \text{se } n = 0 \\ \mathrm{T}_1\left(\left\lfloor \frac{\mathrm{n}}{3} \right\rfloor\right) + \mathrm{n}, \text{se } n > 0 \end{cases}$$

$$\mathrm{T}_2(\mathrm{n}) = \begin{cases} n, \text{se } n = 0, 1, 2 \\ \mathrm{T}_2\left(\left\lfloor \frac{\mathrm{n}}{3} \right\rfloor\right) + \mathrm{T}_2\left(\left\lceil \frac{\mathrm{n}}{3} \right\rceil\right) + \mathrm{n}, \text{se } n > 2 \end{cases}$$

$$\mathrm{T}_3(\mathrm{n}) = \begin{cases} n, \text{se } n = 0, 1, 2 \\ 2 \times \mathrm{T}_3\left(\frac{\mathrm{n}}{3}\right) + \mathrm{n}, \text{se } n \text{ é múltiplo de } 3 \\ \mathrm{T}_3\left(\left\lfloor \frac{\mathrm{n}}{3} \right\rfloor\right) + \mathrm{T}_3\left(\left\lceil \frac{\mathrm{n}}{3} \right\rceil\right) + \mathrm{n}, \text{caso contrário} \end{cases}$$

Deve utilizar **aritmética inteira**: n/3 é igual a $\left\lfloor \frac{\mathrm{n}}{3} \right\rfloor$ e (n+2)/3 é igual a $\left\lceil \frac{\mathrm{n}}{3} \right\rceil$.

- **Preencha a tabela da página seguinte** com o resultado de cada função e o número de chamadas recursivas para os sucessivos valores de n.
- Analisando os dados da tabela, estabeleça uma ordem de complexidade para cada algoritmo?

$\mathcal{O}\left(\left\lfloor \log_3(n) \right\rfloor\right) \to T_1$, $\mathcal{O}(n) \to T_2$ e T_3

- Escreva uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função $\mathbf{T_1(n)}$. Obtenha, depois, uma **expressão exata e simplificada;** determine a sua **ordem de complexidade**. Compare a expressão obtida com a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico**.

Expressão recorrente para as chamadas recursivas:

$$M(n) = \begin{cases} 1 + \left\lfloor M\left(\frac{n}{3}\right)\right\rfloor & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

Resposta: $\begin{cases} 0 & \text{se } n = 0 \\ \left\lfloor \log_3(n) \right\rfloor + 1 & \text{se } n > 0 \end{cases}$

Desenvolvimento telescópico:

$$M(n) = 1 + \left\lfloor M\left(\frac{n}{3}\right)\right\rfloor = 2 + \left\lfloor M\left(\frac{n}{9}\right)\right\rfloor = 3 + \left\lfloor M\left(\frac{n}{27}\right)\right\rfloor = (\ldots) = k + M\left(\frac{n}{3^k}\right) = \left\lfloor \log_3(n) \right\rfloor + M(1) = \left\lfloor \log_3(n) \right\rfloor + 1$$

$$M\left(\frac{n}{3^k}\right) = M(1) \Leftarrow \frac{n}{3^k} = 1 \Leftrightarrow n = 3^k \Leftrightarrow k = \log_3(n)$$

n	$T_1(n)$	Nº de Chamadas Recursivas	$T_2(n)$	Nº de Chamadas Recursivas	$T_3(n)$	Nº de Chamadas Recursivas
0	0	0	0	0	0	0
1	1	1	1	0	1	0
2	2	1	2	0	2	0
3	4	2	5	2	5	1
4	5	2	7	2	7	2
5	6	2	8	2	8	2
6	8	2	10	2	10	1
7	9	2	14	4	14	3
8	10	2	15	4	15	3
9	13	3	19	6	19	2
10	14	3	22	6	22	5
11	15	3	23	6	23	5
12	17	3	26	6	26	3
13	18	3	28	6	28	6
14	19	3	29	6	29	6
15	21	3	31	6	31	3
16	22	3	34	6	34	5
17	23	3	35	6	35	5
18	26	3	38	6	38	2
19	27	3	43	8	43	6
20	28	3	44	8	44	6
21	30	3	49	10	49	4
22	31	3	51	10	51	8
23	32	3	52	10	52	8
24	34	3	54	10	54	4
25	35	3	59	12	59	7
26	36	3	60	12	60	7
27	40	4	65	14	65	3
28	41	4	69	14	69	9

- Escreva uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função **$T_2(n)$. Considere o caso particular $n = 3^k$** e obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

Fórmula recorrente para as chamadas recursivas

$$M(n) = \begin{cases} M\left(\frac{n}{3}\right) + M\left(\frac{n}{3}\right) + 2 & \text{se } n > 2 \\ 0 & \text{se } 0 \leq n \leq 2 \end{cases}$$

Teorema mestre:

$T(n) = a\,T(n/b) + f(n)$

$f(n) = 2$

$a = 2$

$b = 3$

$f(n) \to O(1)$ logo $O(n^d)$

$d = 0$

$a > b^d \iff 2 > 3^0$, logo $O(n^{\log_3 2})$

se $n = 3^k$, $k = \log_3(n)$

$$M(n) = 2 \times M\left(\frac{n}{3}\right) + 2 = \underbrace{2 \times \left(2 \times M\left(\frac{n}{6}\right) + 2\right) + 2}_{6 + 4M\left(\frac{n}{6}\right)} = \underbrace{2 \times \left(6 + 4M\left(\frac{n}{9}\right)\right) + 2}_{14 + 8M\left(\frac{n}{9}\right)} = \underbrace{2 \times \left(14 + 8M\left(\frac{n}{27}\right)\right) + 2}_{30 + 16M\left(\frac{n}{27}\right)} = \left(2^{k+1} - 2\right) + 2^k M\left(\frac{3^k}{3^k}\right)$$

$$= \underbrace{2^{\log_3(n)+1} - 2 + 0}, \iff O\left(n^{\log_3(2)}\right)$$

$\frac{n}{3^k} = 1$

- Pode **generalizar a ordem de complexidade** que acabou de obter para todo o n**? Justifique.**

A função $n^{\log_3(2)}$ é uma Smooth function, logo podemos generalizar para todo o n a complexidade.

- Obtenha uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função $\mathbf{T_3(n)}$.

$$M(n) = \begin{cases} M\left\lceil \frac{n}{3} \right\rceil + M\left\lceil \frac{n+2}{3} \right\rceil + 2 & \text{caso contrário} \\ 1 + M\left(\frac{n}{3}\right) & \text{se } n \text{ multiplo de } 3 \\ 0 & \text{se } 0 \leq n \leq 2 \end{cases}$$

- **Considere o caso particular $n = 3^k$ e** obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

Fórmula recorrente para as chamadas recursivas

$$M(n) = \begin{cases} 1 + M\left(\frac{n}{3}\right) & \text{se } n > 2 \\ 0 & \text{se } 0 \leq n \leq 2 \end{cases}$$

Desenvolvimento:

$\boxed{\text{se } n = 3^k, \; k = \log_3(n)}$

$$M(n) = 1 + M\left(\frac{n}{3}\right) = 2 + M\left(\frac{n}{9}\right) = 3 + M\left(\frac{n}{27}\right) = (\dots) = k + M\left(\frac{3^k}{3^k}\right) = k + M(1) = k = \log_3(n)$$

Teorema Mestre:

$T(n) = aT(n/b) + f(n)$

$f(n) = 1$

$a = 1 \quad 1 = n^d$

$b = 3 \quad (=) \, d = 0$

$a = b^d \rightarrow O(n^d \log n)$

$1 = 3^0$, logo $O(\log(n))$

- Pode **generalizar a ordem de complexidade** que acabou de obter para todo o n? **Justifique.**

A ordem de complexidade que acabamos de obter não generaliza para todo o n, visto que para o caso específico do $n = 3^k$ apenas utilizamos 2 condições das 3 fornecidas, para todo o n tínhamos que considerar todas as condições o que, neste caso, tendo em conta os resultados obtidos em T_2, irá alterar a ordem de complexidade.

- Atendendo às **semelhanças entre $\mathbf{T_2(n)}$ e $\mathbf{T_3(n)}$** estabeleça uma **ordem de complexidade para $\mathbf{T_3(n)}$. Justifique.**

Como uma das condições de $T_3(n)$ é igual a uma condição do $T_2(n)$ podemos estabelecer que a ordem de complexidade do $T_3(n)$ é igual a $T_2(n)$, e como a expressão calculada do $T_2(n)$ é generalizável, $T_3(n)$ tem ordem de complexidade $O(2^{\log_3(n)+1} - 2)$