NOME:	João Diogo Videira Oliveira	**N.º MEC:**	93295

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.

$$T_1(n) = \begin{cases} 0, \text{se } n = 0 \\ T_1\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + n, \text{se } n > 0 \end{cases}$$

$$T_2(n) = \begin{cases} n, \text{se } n = 0, 1, 2 \\ T_2\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + T_2\left(\left\lceil \frac{n}{3} \right\rceil\right) + n, \text{se } n > 2 \end{cases}$$

$$T_3(n) = \begin{cases} n, \text{se } n = 0, 1, 2 \\ 2 \times T_3\left(\frac{n}{3}\right) + n, \text{se } n \text{ é múltiplo de } 3 \\ T_3\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + T_3\left(\left\lceil \frac{n}{3} \right\rceil\right) + n, \text{caso contrário} \end{cases}$$

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

- **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?

$T_1(n) \rightarrow O(\log n)$
$T_2(n) \rightarrow O(n^{\log_y(x)})$
$T_3(\text{n múltiplos de 3}) \rightarrow O(\log n)$
$T_2(\text{n não multiplos de 3}) \rightarrow O(n^{\log_y(x)})$

- 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**.

$$n = 3^k \Longrightarrow \frac{n}{3} = \left\lfloor \frac{n}{3} \right\rfloor$$

Expressão recorrente para o número de chamadas recursivas:

$$B_1(n) = \begin{cases} 0, & se\ n = 0 \\ B_1\left(\frac{n}{3}\right) + 1, & se\ n > 0\ e\ n = 3^k \end{cases}$$

Expressão exata e simplificada:

$$B_1(n) = 1 + B_1\left(\frac{n}{3}\right) = 2 + B_1\left(\frac{n}{9}\right) = 1 + B_1\left(\frac{n}{27}\right) = \cdots = k + B\left(\frac{n}{3^k}\right)$$

$$= k + B_1\left(\frac{3^k}{3^k}\right) = k + B_1(1) = k + 1 = \log_3(n) + 1$$

$$B_1(n) \rightarrow \log_3(n) + 1$$

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	48	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 $\mathbf{T_2(n)}$. **Considere o caso particular** $\boldsymbol{n = 3^k}$ **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**.

$$n = 3^k \Longrightarrow \frac{n}{3} = \left\lfloor\frac{n}{3}\right\rfloor = \left\lceil\frac{n}{3}\right\rceil$$

Expressão recorrente para o número de chamadas recursivas:

$$B_2(n) = \begin{cases} n, & se\ n = 0,1,2 \\ 2B_2\left(\frac{n}{3}\right) + 2, & se\ n > 2\ e\ n = 3^k \end{cases}$$

Expressão exata e simplificada:

$$\begin{aligned} B_2(n) &= 2 + 2B_2\left(\frac{n}{3}\right) = 6 + 4B_2\left(\frac{n}{9}\right) = 14 + 8B_2\left(\frac{n}{27}\right) = \cdots \\ &= 2^{k+1} - 2 + 2^k B_2\left(\frac{n}{3^k}\right) = 2^{k+1} - 2 + 2^k B_2\left(\frac{3^k}{3^k}\right) \\ &= 2^{k+1} - 2 + 2^k B_2(1) = 2^{k+1} - 2 = 2^{\log_3(n)+1} - 2 \end{aligned}$$

$$B_2(n) \to 2^{\log_3(n)+1} - 2$$

Confirmação do resultado utilizando o Teorema Mestre:

$$B(n) = aB\left(\frac{n}{b}\right) + f(n)$$

$$a = 2; b = 3; f(n) = 2; d = 0(f(n) \to O(n^0))$$

$$Como\ a > b^d \to 2 > 3^0, então: B_2(n) \to n^{\log_3 2}$$

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

Sim, visto que, a ordem de complexidade é a mesma para qualquer valor de n.

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

$$B_3(n) = \begin{cases} n, se\ n = 0,1,2 \\ B_3\left(\frac{n}{3}\right) + 1, se\ n\ é\ multiplo\ de\ 3 \\ 2B_3\left(\frac{n}{3}\right) + 2, caso\ contrário \end{cases}$$

- **Considere o caso particular** $\boldsymbol{n = 3^k}$ **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**.

Se n é múltiplo de 3 $(n = 3^k)$:

$$B_3(n) = 1 + B_3\left(\frac{n}{3}\right) = 2 + B_3\left(\frac{n}{9}\right) = 1 + B_3\left(\frac{n}{27}\right) = \cdots = k + B_3\left(\frac{n}{3^k}\right)$$

$$= k + B_3\left(\frac{3^k}{3^k}\right) = k + B_3(1) = k + 1 = \log_3(n) + 1$$

$T_1(n) \rightarrow \log_3(n) + 1$

Confirmação do resultado utilizando o Teorema Mestre:

$B(n) = aB\left(\frac{n}{b}\right) + f(n)$

$a = 1; b = 3; f(n) = 1; d = 0(f(n) \rightarrow O(n^0))$

$Como\ a \geq b^d \rightarrow 1 = 3^0, então: B_3(n) \rightarrow \log_3 n$

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

Não, visto que, a ordem de complexidade difere se o valor for ou não múltiplo de 3, ou seja, se n for múltiplo de 3 a ordem de complexidade é uma, se n não for múltiplo de 3, a ordem de complexidade é outra.

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

Quando n não é múltiplo de 3, $B_3(n)$ é igual a $B_2(n)$.

$B_3(n) \rightarrow\ 2^{\log_3(n)+1} - 2$