AULA 6 - 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.

$$T_1(n) = \begin{cases} 0, se\, n=0 \\ T_1\left(\lfloor \frac{n}{4} \rfloor\right) + n, se\, n>0 \end{cases}$$

$$T_2(n) = \begin{cases} n, se\, n=0,1,2,3 \\ T_2\left(\lfloor \frac{n}{4} \rfloor\right) + T_2\left(\lceil \frac{n}{4} \rceil\right) + n, se\, n>3 \end{cases}$$

$$T_3(n) = \begin{cases} n, se\, n=0,1,2,3 \\ 2 \times T_3\left(\frac{n}{4}\right) + n, se\, n\, é\, múltiplo\, de\, 4 \\ T_3\left(\lfloor \frac{n}{4} \rfloor\right) + T_3\left(\lceil \frac{n}{4} \rceil\right) + n, caso\, contrario \end{cases}$$

Deve utilizar **aritmética inteira**: n/4 é igual a $\lfloor \frac{n}{4} \rfloor$ e (n+3)/4 é igual a $\lceil \frac{n}{4} \rceil$.

- **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.

$$T_1(n) \in \Theta(n)$$
$$T_2(n) \in \Theta\left(n^{(1/2)} \log n\right)$$
$$T_3(n) \in \Theta(n)$$

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

$$C_1(0) = 0$$
$$C_1(n) = 1 + C_1\left(\lfloor \frac{n}{4^k} \rfloor\right) = k + C_1\left(\lfloor \frac{n}{4^k} \rfloor\right)$$
$$Para\, \frac{n}{4^k} = 1 \Leftrightarrow n = 4^k \Leftrightarrow k = \log_4(n)$$
$$Sendo\, assim, temos\, que\, C_1(n) = 1 + \log_4(n)$$
$$Ordem\, de\, complexidade\, será\, de: \Theta(n)$$

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	3	1	3	0	3	0
4	5	2	6	2	6	1
5	6	2	8	2	8	2
6	7	2	9	2	9	2
7	8	2	10	2	10	2
8	10	2	12	2	12	1
9	11	2	14	2	14	2
10	12	2	15	2	15	2
11	13	2	16	2	16	2
12	15	2	18	2	18	1
13	16	2	22	4	22	3
14	17	2	23	4	23	3
15	18	2	24	4	24	3
16	21	3	28	6	28	2
17	22	3	31	6	31	5
18	23	3	32	6	32	5
19	24	3	33	6	33	5
20	26	3	36	6	36	3
21	27	3	38	6	38	6
22	28	3	39	6	39	6
23	29	3	40	6	40	6
24	31	3	42	6	42	3
25	32	3	44	6	44	6
26	33	3	45	6	45	6
27	34	3	46	6	46	6
28	36	3	48	6	48	3

- Escreva uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função $T_2(n)$**. Considere o caso particular** $n=4^k$ **e** obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

$$Expressão\, recorrente: C_2(n)=\begin{cases} 1\, se\, n=0,1,2,3 \\ C_2\left(floor\left(\frac{n}{4}\right)\right)+C_2\left(ceil\left(\frac{n}{4}\right)\right)+2,\, se\, n>3 \end{cases}$$

$$Com\, n=4^k,\, e\, assumindo\, que\, ceil\left(\frac{n}{4}\right)=floor\left(\frac{n}{4}\right)$$

$$C_2(n)=2+2\times C_2\left(\frac{n}{4}\right)=2\times\left(C_2\left(\frac{n}{4^2}\right)+2\right)+2=2^k\times C_2\left(\frac{n}{4^k}\right)+\sum_{i=1}^{k}2^i=2^k\times C_2\left(\frac{n}{4^k}\right)+2^{(i-1)}-2$$

$$...uma\, vez\, que \log_4 n=k,\, a\, complexidade\, é\, linear.$$

$$Com\, a\, expressão\, C_2(n)=2\times C_2\left(\frac{n}{4}\right)+2\, temos:$$

$$a=2,\, b=4\, e\, como\, f(n)\, é\, constante,\, d=0$$

$$A\, partir\, do\, Teorema\, Mestre\, podemos\, afirmar\, que\, a>b^d \Leftrightarrow 2>1\, resultando\, em\, C_2(n)=\Theta\left(n^{(\log_4 2)}\right)$$

$$Confirmando\, assim\, a\, complexidade\, linear\, O(n).$$

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

Sim podemos, pois o mesmo pois o comportamento da função não se altera para diferentes n's

- Obtenha uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função $T_2(n)$.

$$Expressão\, recorrente: C_2(n)=\begin{cases} 1\, se\, n=0,1,2,3 \\ C_2\left(floor\left(\frac{n}{4}\right)\right)+C_2\left(ceil\left(\frac{n}{4}\right)\right)+2,\, se\, n>3 \end{cases}$$

- **Considere o caso particular** $n = 4^k$ **e** obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

$$C_3(n) = \begin{cases} 1, \text{ se } n = 0,1,2,3 \\ 1 + C_3\left(\left\lfloor \frac{n}{4} \right\rfloor\right), \text{ se } n > 3 \text{ e mult. de } 4 \\ C_3\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + C_3\left(\left\lceil \frac{n}{4} \right\rceil\right) + 2, \text{ se } n > 3 \text{ e ñ mult. de } 4 \end{cases}$$

Considerando $n = 4^k$ e $\left\lceil \frac{n}{4} \right\rceil = \left\lfloor \frac{n}{4} \right\rfloor$

- Para múltiplos de 4
  - $C_3'(n) = 1 + C_3'\left(\frac{n}{4}\right) \; \ldots \; C_3'(4) = 1 + C_3'(1) = 2 \; \ldots \; C_3'(n) = k + C_3'\left(\frac{n}{4^k}\right)$

    e como $n = 4^k \Leftrightarrow \boxed{k = \log_4 n} \Leftrightarrow \boxed{\frac{n}{4^k} = 1}$

    $C_3'(n) = \log_4 n + C_3'(1) \Leftrightarrow C_3'(n) = 1 + \log_4 n$

  ... assim, pelo Teorema Mestre

  $C_3'(n) = \Theta(n)$

  Resultando em complexidade linear

- Para não múltiplos de 4
  - $C_3''(n) = 2 + C_3''\left(\frac{n}{4}\right) + C_3''\left(\frac{n}{4}\right) = 2 + 2 \cdot C_3''\left(\frac{n}{4}\right)$

    $= 2 + 2 \cdot \left(2 + 2 \cdot C_3''\left(\frac{n}{4^2}\right)\right) = \ldots$

    $= 2^k \cdot C_3''\left(\frac{n}{4^k}\right) + \underbrace{\sum_{i=1}^{k} 2^i}$

    $= 2^k \cdot C_3''\left(\frac{n}{4^k}\right) + 2^{k-1} - 2$

    $= 2^{\log_4 n} \cdot C_3''\left(\frac{n}{4^{\log_4 n}}\right) + 2^{\log_4 n} - 2$

  com $n = 4^k$

  $\log_4 n = k$

  - $\sum_{i=1}^{k} 2^i = 2^{k-1} - 2$

    razão: $\frac{a_{i+1}}{a_i} = \frac{2^{i+1}}{2^i} = 2$

    $a_1 = 2$, assim:

    $2 \times \frac{1 - 2^i}{1 - 2} = 2 - 2^{i+1}$

  ... Com o uso do Teorema Mestre

  $a > b^d \Leftrightarrow 2 > 4^0 \Leftrightarrow 2 > 1$

  $C_2(n) = \Theta(n^{\log_4 2}) = \Theta(n^{1/2})$

  Podemos deduzir então que é de complexidade linear.

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

Não, pois o mesmo pois o comportamento da função e subsequentemente a sua complexidade são diferentes para <u>*n's*</u> múltiplos de 4 e para o que não o são.

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

Visto que $T_3(n)$ é como que uma composição de $T_1(n)$ para múltiplos de 4, e de $T_2(n)$ para não múltiplos de 4, e estes dois apresentam complexidade linear, também $T_3(n)$ é de complexidade lineae