NOME:	MIGUEL JOSÉ FERREIRA CABRAL	N.º MEC:	93091

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 análise 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 $\lfloor \frac{n}{3} \rfloor$ e (n+2)/3 é igual a $\lceil \frac{n}{3} \rceil$.

- **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?

T1 tem ordem de complexidade logarítmica → O(logn)
T2 tem ordem de complexidade linear → O(n)
T3 tem ordem de complexidade linear →O(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 a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico**.

$C(0) = 0$ →caso inicial

$C(n) = C(floor(n/3) + 1 = C(floor(n/9) + 2 = C(floor(n/3^k) + k$

Para n = 0 , k = $1 + log_3 n$

$C(n) = C(floor(n/3^{1+log_3 n}) + 1 + log_3 n$ = C(0) + 1 + $log_3 n$

De acordo com a expressão obtida podemos concluir que tem ordem de complexidade logarítmica tal como os resultados obtidos experimentalmente.

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**.

C(0) = 0 , C(1) = 0, C(2) = 0 → Casos iniciais
C(n) = C(floor(n/3)) + C(ceil(n/3)) + 2

Para n = $3^k \Leftrightarrow k = log_3 n$ temos que → C(n) = 2*C(n/3) + 2 = 2*(2*C(n/9) +2) + 2 = 2*(6+4*C(n/27)) + 2 = $2^{k+1} - 2$, logo C(n) = $2^{log_3 n+1} - 2 = 2 * n^{log_3 2} - 2$ → O($n^{log_3 2}$)
Teorema Mestre:
a = 2; b = 3 ; d = 0 como $a > b^d$ → O($n^{log_3 2}$)
Confirmando o resultado obtido anteriormente.

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

Sim. Como o desenvolvimento telescópico da função é do tipo $n^{\alpha}(\alpha > 0)$ é uma função suave , logo a ordem de complexidade pode ser generalizada para todo o n.

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

C(0) = 0 , C(1) = 0, C(2) = 0 → Casos iniciais
C(n) = 1 + C(n/3) , se n é multiplo de 3
C(n) = C(n/3) + C((c+2)/3) +2 , caso contrário

- **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**.

Para $n = 3^k$ **, n** ou é 0 ou múltiplo de 3 logo só vamos analizar estes casos.
C(0) = 0 , C(1) = 0, C(2) = 0 → Casos iniciais
Para n multiplo de 3 :
C(n) = 1 + C(n/3) = 2 + C(n/9) = 3 + C(n/27) = k + C(n/ 3^k) , substituindo k por $log_3 n$:
C(n) = $log_3 n$ + C(n/ $3^{log_3 n}$) = $log_3 n$ + C(1) = $log_3 n$ → O($log_3 n$)
Teorema Mestre:
a = 1; b = 3 ; d = 0 como a = b^d → O($n^d \; log_b n$) → O($log_3 n$)
Confirmando o resultado obtido anteriormente.

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

Sim. Como o desenvolvimento telescópico da função é do tipo $log_{\alpha}n$ $(\alpha > 0)$ é uma função suave , logo a ordem de complexidade pode ser generalizada para todo o n.

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

Uma vez que calculam o mesmo resultado mas T3 faz menos operações a ordem de complexidade de T3 não pode ser superior à ordem de complexidade de T2.