

NOME:	JOÃO PEDRO COSTA GAMEIRO	N.º MEC:	93097
-------	--------------------------	----------	-------

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.

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

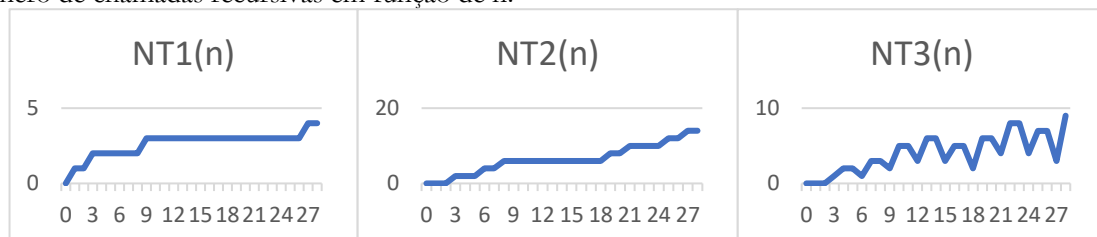
$$T_2(n) = \begin{cases} n, & \text{se } n = 0, 1, 2 \\ T_2\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + T_2\left(\left\lceil \frac{n}{3} \right\rceil\right) + n, & \text{se } n > 2 \end{cases}$$

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

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?

Para analisar os valores da tabela, vamos analisar três gráficos (um para cada algoritmo) que especificam o valor do número de chamadas recursivas em função de n .



Ao observarmos o gráfico $NT1(n)$ podemos concluir que o número de chamadas evolui logaritmicamente, o que leva à conclusão de que a ordem de complexidade para o primeiro algoritmo é logarítmica, $O(\log n)$.

No caso $NT2(n)$, o valor das chamadas recursivas já cresce linearmente, o que nos leva a afirmar que a complexidade do segundo algoritmo é linear, $O(n)$.

O terceiro gráfico tal como o segundo revela uma evolução linear, no entanto apresenta frequentes oscilações, o que dificulta a determinação da complexidade. Contudo não é errado considerar que a complexidade do terceiro algoritmo é 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**.

Expressão recorrente:
$$N_{T_1}(n) = \begin{cases} 0, & x = 0 \\ N\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + 1, & n > 0, \end{cases} \quad x > 0$$

Agora vamos calcular uma fórmula fechada para o número de chamadas recursivas de $T_1(n)$ ($N_{T_1}(n)$), utilizando o desenvolvimento telescópico:

$$N_{T_1}(n) = N_{T_1}\left(\frac{n}{3}\right) + 1 = N_{T_1}\left(\frac{n}{3^2}\right) + 2 = N_{T_1}\left(\frac{n}{3^3}\right) + 3 = N_{T_1}\left(\frac{n}{3^k}\right) + k$$

Se escolhermos um k tal que $\frac{n}{3^k} = 1$, para obter os casos base vamos ter que $\log_3 n = k$ o que leva a que:

$$N_{T_1}(n) = N_{T_1}\left(\frac{n}{3^{\log_3 n}}\right) + \log_3 n$$

E daqui podemos concluir que a ordem de complexidade é logarítmica, $O(\log n)$, tal como a obtida por observação dos valores da tabela.

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	4	10	1
7	9	2	14	4	14	3
8	10	2	15	6	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**.

Expressão recorrente:
$$N_{T_2}(n) = \begin{cases} 0, & n \leq 2 \\ N_{T_2}\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + N_{T_2}\left(\left\lceil \frac{n}{3} \right\rceil\right) + 2, & n > 2 \end{cases}$$

Se considerarmos o caso particular em que $n = 3^k$ vamos obter que para este valor $\left\lfloor \frac{n}{3} \right\rfloor = \left\lceil \frac{n}{3} \right\rceil = \frac{n}{3}$. Assim sendo, usando o desenvolvimento telescópico:

$$\begin{aligned} N_{T_2}(n) &= N_{T_2}\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + N_{T_2}\left(\left\lceil \frac{n}{3} \right\rceil\right) + 2 = 2 \cdot N_{T_2}\left(\frac{n}{3}\right) + 2 = 2 \cdot \left(2 \cdot N_{T_2}\left(\frac{n}{3^2}\right) + 2\right) + 2 = \\ &= 2 \cdot \left(2 \cdot \left(2 \cdot N_{T_2}\left(\frac{n}{3^k}\right) + 2\right) + 2\right) + 2 = 2^k \cdot N_{T_2}\left(\frac{n}{3^k}\right) + \sum_{i=1}^k 2^i = 2^k \cdot N_{T_2}\left(\frac{n}{3^k}\right) + 2^{k+1} - 2 \end{aligned}$$

revela uma complexidade linear pois $\log_3 n = k$.

Assim sendo se considerarmos a expressão $N_{T_2}(n) = 2 \cdot N_{T_2}\left(\frac{n}{3}\right) + 2$ obtida anteriormente, em que $a = 2, b = 3$ e $f(n) = 2$, podemos concluir que $f(n) = \Theta(n^d)$ e que $d=0$. Nesta expressão a representa o número de instâncias menores e b o fator constante.

Logo aplicando o teorema mestre, podemos concluir que:

$a > b^d \Leftrightarrow 2 > 3^0$ e concluímos $N_{T_2}(n) = \Theta(n^{\log_3 2})$, o que revela uma complexidade linear, $O(n)$ que foi também obtida por observação dos valores da tabela e no desenvolvimento telescópico.

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

Sabemos que a expressão $N_{T_2}(n) = 2 \cdot N_{T_2}\left(\frac{n}{3}\right) + 2$ é eventualmente não-decrescente, pois

$$N_{T_2}(n_1) \leq N_{T_2}(n_2), \forall n_1, n_2 \text{ tal que } n_2 > n_1 \geq n_0.$$

Na alínea anterior vimos que $N_{T_2}(n) = \Theta(n^{\log_3 2})$, para todos os valores de n que são potências de 3 ($n = 3^k$). Como $n^{\log_3 2}$ é uma função suave e $N_{T_2}(n) = \Theta(n^{\log_3 2})$, podemos generalizar a ordem de complexidade obtida para todo o n .

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

$$\text{Expressão recorrente: } N_{T_3}(n) = \begin{cases} 0 & , n \leq 2 \\ N_{T_3}\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + 1 & , n \text{ é múltiplo de } 3 \\ N_{T_3}\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + N_{T_3}\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + 2 & , n \text{ caso contrário} \end{cases}$$

- 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**.

Todos os valores para os quais $n = 3^k$, são múltiplos de 3. Logo vamos aplicar o desenvolvimento telescópico partindo da expressão do segundo ramo da fórmula recorrente:

$$N_{T_3}(n) = N_{T_3}\left(\frac{n}{3}\right) + 1 = N_{T_3}\left(\frac{n}{3^2}\right) + 2 = N_{T_3}\left(\frac{n}{3^3}\right) + 3 = N_{T_3}\left(\frac{n}{3^k}\right) + k \quad \text{revela uma complexidade logarítmica pois } \log_3 n = k.$$

Aplicando então agora o teorema mestre à fórmula $N_{T_3}(n) = N_{T_3}\left(\frac{n}{3}\right) + 1$ em que $a = 1, b = 3$ e $f(n) = 1$.

Sabendo que $f(n) = \Theta(n^d)$ e que $d=0$, podemos concluir que

$a = b^d \Leftrightarrow 1 = 3^0$ e assim afirmar que $N_{T_3}(n) = \Theta(n^0 \log n) = \Theta(\log n)$, que é o mesmo valor obtido pelo desenvolvimento telescópico.

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

Neste caso não podemos utilizar a regra da “suavidade” pois o algoritmo não nos revela um número de chamadas recursivas que eventualmente seja não decrescente, contrariamente revela um número de chamadas recursivas crescente, com oscilações sempre que n é múltiplo de 3 (ver gráfico da primeira questão ($T_3(n)$)). O valor que obtivemos para a alínea anterior é apenas válido para $n = 3^k$ e não podemos por isso generalizar a ordem de complexidade 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**.

Ao observar os valores obtidos para o número de chamadas recursivas dos algoritmos $T_2(n)$ e $T_3(n)$, concluímos que são bastante semelhantes. No entanto os valores obtidos por $T_3(n)$, nunca chegam a ser maiores do que os obtidos por $T_2(n)$, ou seja $T_2(n)$ majora $T_3(n)$ em termos de esforço computacional.

Concluindo, como $T_3(n)$, é majorado por $T_2(n)$ significa que a sua ordem de complexidade pode tomar valores inferiores ou até mesmo iguais, mas nunca maiores. Ou seja, não é errado afirmar que a ordem de complexidade de $T_3(n)$ é igual à de $T_2(n)$.