

GUIÃO 05 – ANÁLISE DA COMPLEXIDADE DE ALGORITMOS RECURSIVOS

Implemente os seguintes **algoritmos recursivos**, sem recorrer a funções de arredondamento (**floor** e **ceil**). Note que, considerando o quociente da divisão inteira, temos que $n/2$ é igual a $\lfloor \frac{n}{2} \rfloor$ e $(n+1)/2$ é igual a $\lceil \frac{n}{2} \rceil$.

Determine o **número de chamadas recursivas** executadas por cada função.

- $T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n, & \text{se } n > 1 \end{cases}$
- $T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n, & \text{se } n > 1 \end{cases}$
- $T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + n, & \text{se } n \text{ é ímpar} \\ 2 \times T\left(\frac{n}{2}\right) + n, & \text{se } n \text{ é par} \end{cases}$

Os números das chamadas são menos um do que esta escrito porque só queriam as chamadas recursivas e não o número total de chamadas

Preencha a tabela, com o valor do resultado da função e o número de chamadas recursivas efetuadas, para os sucessivos valores de n , por exemplo, até 32 ou 64 ou 128.

N	1ª Função (N)	Nº de Chamadas	2ª Função (N)	Nº de Chamadas	3ª Função (N)	Nº de Chamadas recursivas
1	1	1	1	1	1	1
2	3	2	4	3	4	2
3	4	2	8	5	8	4
4	7	3	12	7	12	3
5	8	3	17	9	17	7
6	10	3	22	11	22	5
7	11	3	27	13	27	8
8	15	4	32	15	32	4
9	16	4	38	17	38	11
10	18	4	44	19	44	8
11	19	4	50	21	50	13
12	22	4	56	23	56	6
13	23	4	62	25	62	14
14	25	4	68	27	68	9
15	26	4	74	29	74	13

Analisando os dados da tabela, qual é a **ordem de complexidade** de cada algoritmo?

Complexidade T1 é $O(\log n)$.
Complexidade T2 é $O(2n-2)$
T3 par $O(\log n)$; ímp. $O(N)$

Determine formalmente a **ordem de complexidade dos dois primeiros algoritmos**, obtendo

expressões matemáticas exatas e simplificadas. Complexidade T1 é $O(\log n)$. Complexidade T2 é $O(2n-2)$

No caso do **terceiro algoritmo** indique para que valores de N se obtém o **melhor e o pior caso** e faça a respetiva análise da complexidade.

A complexidade é entre $O(\log n)$ para números pares e $O(n)$ para números ímpares. Assim sendo o pior caso seria um valor com o maior número de divisões ímpares, enquanto que o melhor caso seria igual a 0. E depois continuaria a ser melhor sempre que fosse dividido e desse par. ¹