

AULA 7 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS RECURSIVOS (ALGORITMOS SIMPLES)

Implemente os seguintes **algoritmos recursivos** – sem recorrer a funções de arredondamento (floor e ceil) – e determine o **número de invocações recursivas** executadas por cada algoritmo:

$$\bullet \quad T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n, & \text{se } n > 1 \end{cases}$$

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

Escreva o código das funções no verso da folha

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

Tenha em atenção que nas linguagens de programação a divisão inteira atua por truncatura. Assim sendo $n/2$ é igual a $\left\lfloor \frac{n}{2} \right\rfloor$ e $(n+1)/2$ é igual a $\left\lceil \frac{n}{2} \right\rceil$.

- Preencha a tabela com o valor da função e o número de invocações recursivas para os sucessivos valores de n .

N	1ª Função (N)	Nº de Invocações	2ª Função (N)	Nº de Invocações	3ª Função (N)	Nº de Invocações
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
$O(N)$						

- Analisando os dados da tabela qual é a ordem de complexidade de cada algoritmo?
- Determine formalmente a ordem de complexidade dos dois primeiros algoritmos, obtendo expressões matemáticas exatas e simplificadas.
- No caso do terceiro algoritmo indique para que valores de N se obtém o melhor e o pior casos e faça as respetivas análises.

