

## AULA 4 – ANÁLISE DA COMPLEXIDADE DE ALGORITMOS

**\*\*\* Entregue, num ficheiro ZIP, este guião preenchido e o código desenvolvido \*\*\***

1 - Seja uma dada sequência (*array*) de  $n$  elementos inteiros e não ordenada. Pretende-se determinar quantos elementos da sequência respeitam a seguinte propriedade:

$$\text{array}[i] = \text{array}[i - 1] + \text{array}[i + 1], \text{ para } 0 < i < (n - 1)$$

- Implemente uma **função eficiente e eficaz** que determine quantos elementos (resultado da função) de uma sequência com  $n$  elementos (sendo  $n > 2$ ) respeitam esta propriedade.

**Depois de validar o algoritmo apresente a função no verso da folha.**

- Pretende-se determinar experimentalmente a **ordem de complexidade do número de comparações** efetuadas pelo algoritmo e envolvendo elementos da sequência.
- Considere as seguintes sequências de 10 elementos inteiros, que cobrem algumas situações possíveis de execução do algoritmo.  
Determine, para cada uma delas, o número de elementos que obedecem à condição e o número de comparações efetuadas, envolvendo elementos da sequência.

1	2	3	4	5	6	7	8	9	10
1	2	1	4	5	6	7	8	9	10
1	2	1	3	2	6	7	8	9	10
0	2	2	0	3	3	0	4	4	0
0	0	0	0	0	0	0	0	0	0

Resultado	
Resultado	
Resultado	
Resultado	
Resultado	

Nº de operações	
Nº de operações	
Nº de operações	
Nº de operações	
Nº de operações	

**Depois dos testes experimentais responda às seguintes questões:**

- Em termos do número de comparações efetuadas, podemos distinguir alguma variação na execução do algoritmo? Ou seja, existe a situação de melhor caso e de pior caso, ou estamos perante um algoritmo com caso sistemático?

- Com base nos resultados experimentais, qual é a ordem de complexidade do algoritmo? Justifique.

- Determine formalmente a ordem de complexidade do algoritmo. Tenha em atenção que deve obter uma expressão matemática exata e simplificada.

**Faça a análise no verso da folha.**

- Calcule o valor da expressão para  $n = 10$  e compare-o com os resultados obtidos experimentalmente.

NOME:

Nº MEC:

**FUNÇÃO**

**ANÁLISE FORMAL DO ALGORITMO**

NOME:

Nº MEC:

2 - Seja uma dada sequência (*array*) de  $n$  elementos inteiros e não ordenada. Pretende-se determinar quantos ternos  $(i, j, k)$  de índices da sequência respeitam a seguinte propriedade:

$$\text{array}[k] = \text{array}[i] + \text{array}[j], \text{ para } i < j < k$$

- Implemente uma **função eficiente e eficaz** que determine quantos ternos  $(i, j, k)$  de índices (resultado da função) de uma sequência com  $n$  elementos (sendo  $n > 2$ ) respeitam esta propriedade. **Depois de validar o algoritmo apresente a função no verso da folha.**
- Pretende-se determinar experimentalmente a **ordem de complexidade do número de comparações** efetuadas pelo algoritmo e envolvendo elementos da sequência.
- Considere as sequências anteriormente indicadas de 10 elementos inteiros e outras sequências diferentes à sua escolha; **use sequências com 5, 10, 20, 30 e 40 elementos**. Determine, para cada uma delas, quantos ternos  $(i, j, k)$  de índices respeitam propriedade e o número de comparações efetuadas.

**Depois dos testes experimentais responda às seguintes questões:**

- Em termos do número de comparações efetuadas, podemos distinguir alguma variação na execução do algoritmo? Ou seja, existe a situação de melhor caso e de pior caso, ou estamos perante um algoritmo com caso sistemático?

- Com base nos resultados experimentais, qual é a ordem de complexidade do algoritmo? Justifique.

- Determine formalmente a ordem de complexidade do algoritmo. Tenha em atenção que deve obter uma expressão matemática exata e simplificada.

**Faça a análise no verso da folha.**

- Calcule o valor da expressão para  $n = 10$  e compare-o com os resultados obtidos experimentalmente.

**FUNÇÃO****ANÁLISE FORMAL DO ALGORITMO**

NOME:

Nº MEC: