

AULA 3 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS

Seja uma dada sequência (*array*) de números inteiros não ordenada. Pretende-se determinar quantos elementos da sequência são diferentes dos elementos anteriores. Ou seja:

$$\text{array}[i] \neq \text{array}[i-1], \text{ para } i > 0$$

- Implemente uma função inteira **eficiente** e **eficaz** que determina quantos elementos (resultado da função) de uma sequência com n elementos (sendo $n > 1$) respeitam esta propriedade.
Depois de validar o algoritmo apresente-o no verso da folha

- Determine experimentalmente a **complexidade do número de comparações** efectuadas envolvendo elementos da sequência. Considere as seguintes dez sequências de dez inteiros todas diferentes e que cobrem todas as situações possíveis distintas de execução do algoritmo. Calcule para cada uma delas o número de elementos que obedecem à condição e o número de comparações executadas.

3	3	3	3	3	3	3	3	3	3	Resultado		Nº de operações	
4	3	3	3	3	3	3	3	3	3	Resultado		Nº de operações	
4	5	3	3	3	3	3	3	3	3	Resultado		Nº de operações	
4	5	1	3	3	3	3	3	3	3	Resultado		Nº de operações	
4	5	1	2	3	3	3	3	3	3	Resultado		Nº de operações	
4	5	1	2	6	3	3	3	3	3	Resultado		Nº de operações	
4	5	1	2	6	8	3	3	3	3	Resultado		Nº de operações	
4	5	1	2	6	8	7	3	3	3	Resultado		Nº de operações	
4	5	1	2	6	8	7	9	3	3	Resultado		Nº de operações	
4	5	1	2	6	8	7	9	3	0	Resultado		Nº de operações	

Depois da execução do algoritmo responda às seguintes questões:

- Em termos do número de comparações efectuadas 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?
- Qual é a ordem de complexidade do algoritmo?
- Determine formalmente a ordem de complexidade do algoritmo. Tenha em atenção que deve obter uma expressão matemática exacta 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.

APRESENTAÇÃO DO ALGORITMO**ANÁLISE FORMAL DO ALGORITMO**

E(N) =

NOME:

Nº MEC:


Seja uma dada sequência (*array*) de números inteiros não ordenada. Pretende-se determinar qual é o primeiro elemento da sequência que tem mais elementos menores do que ele atrás de si, indicando a posição (índice do array) onde ele se encontra.

Por exemplo, na sequência { 1, 9, 2, 8, 3, 4, 5, 3, 7, 2 } o 7 que está no índice oito da sequência é maior do que seis elementos seus predecessores. Na sequência { 1, 7, 4, 6, 5, 2, 3, 2, 1, 0 } o 6 que está no índice três da sequência é maior do que dois elementos seus predecessores. Mas, na sequência { 2, 2, 2, 2, 2, 2, 2, 2, 2 } nenhum elemento é maior do que os seus predecessores, pelo que deve ser indicada a posição -1 como resultado.

- Implemente uma função inteira **eficiente** e **eficaz** que determina a posição do primeiro elemento da sequência (resultado da função) de uma sequência com n elementos (sendo $n > 1$) que tem mais predecessores menores do que ele. Depois de validar o algoritmo apresente-o no verso da folha
- Determine experimentalmente a **complexidade do número de comparações** efectuadas envolvendo elementos da sequência. Considere as sequências anteriormente indicadas de dez inteiros e outras sequências diferentes à sua escolha. Calcule para cada uma delas a posição do elemento e o número de comparações executadas.

Depois da execução do algoritmo responda às seguintes questões:

- Em termos do número de comparações efectuadas 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?
- Qual é a ordem de complexidade do algoritmo?
- Determine formalmente a ordem de complexidade do algoritmo. Tenha em atenção que deve obter uma expressão matemática exacta 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.

APRESENTAÇÃO DO ALGORITMO**ANÁLISE FORMAL DO ALGORITMO**

E(N) =

NOME:

Nº MEC: