

NMEC:	93091	NOME:	MIGUEL JOSÉ FERREIRA CABRAL
-------	-------	-------	-----------------------------

AULA 3 – ANÁLISE DA COMPLEXIDADE DE ALGORITMOS

1 - Seja uma dada sequência (*array*) de n elementos inteiros. Pretende-se determinar quantos elementos da sequência são diferentes do seu elemento anterior. Ou seja:

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

Implemente uma **função eficiente e eficaz** que determine 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 **ordem de complexidade do número de comparações** efetuadas pelo algoritmo e envolvendo elementos da sequência. Considere as seguintes 10 sequências de 10 elementos inteiros, todas diferentes, e que cobrem as distintas 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.

Sequência	Resultado	N.º de operações
{3, 3, 3, 3, 3, 3, 3, 3, 3, 3}	0	9
{4, 3, 3, 3, 3, 3, 3, 3, 3, 3}	1	9
{4, 5, 3, 3, 3, 3, 3, 3, 3, 3}	2	9
{4, 5, 1, 3, 3, 3, 3, 3, 3, 3}	3	9
{4, 5, 1, 2, 3, 3, 3, 3, 3, 3}	4	9
{4, 5, 1, 2, 6, 3, 3, 3, 3, 3}	5	9
{4, 5, 1, 2, 6, 8, 3, 3, 3, 3}	6	9
{4, 5, 1, 2, 6, 8, 7, 3, 3, 3}	7	9
{4, 5, 1, 2, 6, 8, 7, 9, 3, 3}	8	9
{4, 5, 1, 2, 6, 8, 7, 9, 3, 0}	9	9

Depois da execução do algoritmo 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?

Estamos perante um algoritmo com caso sistemático, logo não existe nenhuma situação de melhor e de pior caso.

Qual é a ordem de complexidade do algoritmo?

$E(N) \in O(N)$ ou seja linear.

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.

$E(10) = 10 - 1 = 9$, como foi obtido experimentalmente.

APRESENTAÇÃO DO ALGORITMO

```
unsigned int nComp = 0;

unsigned int countDif(int array[],int n){
    int ant = array[0];
    int count = 0;
    for(int i = 1; i < n;i++){
        if(array[i] != ant ){
            count;
        }
        ant = array[i];
        nComp++;
    }
    return count;
}
```

ANÁLISE FORMAL DO ALGORITMO

$$E(N) = \sum_{i=1}^{n-1} 1 = n - 1 - 1 + 1 = n - 1$$

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

Por exemplo, na sequência {1, 9, 2, 8, 3, 4, 5, 3, 7, 2} o elemento 7, que está na posição de índice 8 da sequência, **é maior do que** 6 elementos seus predecessores. Na sequência {1, 7, 4, 6, 5, 2, 3, 2, 1, 0} o elemento 6, que está na posição de índice 3 da sequência, **é maior do que** 2 elementos seus predecessores. Mas, na sequência {2, 2, 2, 2, 2, 2, 2, 2, 2, 2} nenhum elemento é maior do que qualquer um dos seus predecessores, pelo que deve ser devolvido -1 como resultado.

Implemente uma **função eficiente e eficaz** que determine o índice do primeiro elemento (resultado da função) de uma sequência com n elementos (sendo $n > 1$) que tem o maior número de predecessores menores do que ele.

Depois de validar o algoritmo apresente-o no verso da folha.

Determine experimentalmente a **ordem de complexidade do número de comparações** efetuadas envolvendo elementos da sequência. Considere as sequências anteriormente indicadas de 10 elementos inteiros e outras sequências diferentes à sua escolha. Determine, para cada uma delas, o índice do elemento procurado e o número de comparações efetuadas.

Sequência	Resultado(índice)	Nº de comparações
{1, 9, 2, 8, 3, 4, 5, 3, 7, 2}	8	45
{1, 7, 4, 6, 5, 2, 3, 2, 1, 0}	3	45
{2, 2, 2, 2, 2, 2, 2, 2, 2, 2}	-1	45
{2, 2, 2, 2, 5}	4	10

Depois da execução do algoritmo 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?

Estamos perante um algoritmo com caso sistemático, logo não existe situação de melhor e pior caso.

Qual é a ordem de complexidade do algoritmo?

$E(N) \in O(n^2)$, ou seja quadrática.

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.

$$E(10) = 10 * \frac{(10-1)}{2} = 45, \text{ como foi obtido experimentalmente.}$$

APRESENTAÇÃO DO ALGORITMO

```
unsigned int nComp = 0;

int findGreatest(int array[],int n){
    int count = 0;          // variável que conta o número de elementos menores
    int p_count = 0;        // variável que guarda a última maior contagem
    int greatest = -1;       // indice do elemento com maiores elementos
    //menores que ele

    for(int i = 1; i < n;i++){
        for(int j = 0;j < i;j++){
            if(array[i] > array[j]){
                count++;
            }
            nComp ++;
        }
        if(count > p_count){
            greatest = i;
            p_count = count;
        }
        count = 0;
    }
    return greatest;
}
```

ANÁLISE FORMAL DO ALGORITMO

$$\mathbf{E}(\mathbf{N}) = \sum_{i=1}^{n-1} \left(\sum_{j=0}^{n-1} 1 \right) = \sum_{i=1}^{n-1} n = n * \frac{(n-1)}{2}$$