

NOME:	RODRIGO LOPES MARTINS	N.º MEC:	93264
--------------	------------------------------	-----------------	--------------

AULA 4 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS

1 - Considere uma sequência (*array*) de n elementos inteiros, ordenada por **ordem não decrescente**. Pretende-se determinar se a sequência é uma **progressão aritmética de razão 1**, i.e., $a[i+1] - a[i] = 1$.

- Implemente uma função **eficiente** (utilize um algoritmo em lógica negativa) e **eficaz** que verifique se uma sequência com n elementos ($n > 1$) define uma sequência contínua de números. A função deverá devolver 1 ou 0, consoante a sequência verificar ou não essa propriedade. **Depois de validar o algoritmo apresente-o no verso da folha.**
- Determine experimentalmente a **ordem de complexidade do número de adições/subtrações** 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, se satisfaz a propriedade e qual o número de operações de adição/subtração efetuadas pelo algoritmo.

Sequência	Resultado	N.º de operações
{1, 3, 4, 5, 5, 6, 7, 7, 8, 9},	0	1
{1, 2, 4, 5, 5, 6, 7, 8, 8, 9},	0	2
{1, 2, 3, 6, 8, 8, 8, 9, 9, 9},	0	3
{1, 2, 3, 4, 6, 7, 7, 8, 8, 9},	0	4
{1, 2, 3, 4, 5, 7, 7, 8, 8, 9},	0	5
{1, 2, 3, 4, 5, 6, 8, 8, 9, 9},	0	6
{1, 2, 3, 4, 5, 6, 7, 9, 9, 9},	0	7
{1, 2, 3, 4, 5, 6, 7, 8, 8, 9},	0	8
{1, 2, 3, 4, 5, 6, 7, 8, 9, 9},	0	9
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	1	9

Depois da execução do algoritmo responda às seguintes questões:

- Qual é a sequência (ou as sequências) que corresponde(m) ao melhor caso do algoritmo?

A primeira sequência corresponde ao melhor caso do algoritmo.

- Qual é a sequência (ou as sequências) que corresponde(m) ao pior caso do algoritmo?

A última e a penúltima sequências são as que correspondem ao pior caso do algoritmo.

- Determine o número de adições efetuadas no caso médio do algoritmo (**para $n = 10$**).

São efetuadas 5 adições no caso médio.

- Qual é a ordem de complexidade do algoritmo?

O algoritmo é de complexidade linear $O(n)$.

- Determine formalmente a ordem de complexidade do algoritmo nas situações do melhor caso, do pior caso e do caso médio, considerando uma sequência de tamanho n . Tenha em atenção que deve obter expressões matemáticas exatas e simplificadas.

ANÁLISE FORMAL DO ALGORITMO
MELHOR CASO - $B(N) = 1$
PIOR CASO - $W(N) = N-1$
CASO MÉDIO - $A(N) = N/2$

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

$B(10)=1$
 $W(10)=9$
 $A(10)=5$

Os valores calculados correspondem aos resultados experimentais.

APRESENTAÇÃO DO ALGORITMO
<pre> int operacoes = 0; int isValid(int* a,int n){ assert(n > 1); int i; for (i = 0; i < n-1; i++) { operacoes++; if (a[i+1] - a[i] != 1) { return 0; } } return 1; } </pre>

2 - Considere uma sequência (array) não ordenada de n elementos inteiros. Pretende-se eliminar os elementos repetidos existentes na sequência, sem fazer uma pré-ordenação e sem alterar a posição relativa dos elementos. Por exemplo, a sequência { 1, 2, 2, 2, 3, 3, 4, 5, 8, 8 } com 10 elementos será transformada na sequência { 1, 2, 3, 4, 5, 8 } com apenas 6 elementos. Por exemplo, a sequência { 1, 2, 2, 2, 3, 3, 3, 3, 8, 8 } com 10 elementos será transformada na sequência { 1, 2, 3, 8 } com apenas 4 elementos. Por exemplo, a sequência { 1, 2, 3, 2, 1, 3, 4 } com 7 elementos será transformada na sequência { 1, 2, 3, 4 } com apenas 4 elementos. Mas, a sequência { 1, 2, 5, 4, 7, 0, 3, 9, 6, 8 } permanece inalterada.

- Implemente uma função **eficiente** e **eficaz** que elimina os elementos repetidos numa sequência com n elementos ($n > 1$). A função deverá ser *void* e alterar o valor do parâmetro indicador do número de elementos efetivamente armazenados na sequência (que deve ser passado por referência).

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

- Determine experimentalmente a **ordem de complexidade do número de comparações** e **do número de deslocamentos** envolvendo elementos da sequência. Considere as sequências anteriormente indicadas de 10 elementos e outras à sua escolha. Determine, para cada uma delas, a sua configuração final, bem como o número de comparações e de deslocamentos efetuados.

Depois da execução do algoritmo responda às seguintes questões:

- Indique uma sequência inicial com 10 elementos que conduza ao **melhor caso do número de comparações** efetuadas. Qual é a sequência final obtida? Qual é o número de comparações efetuadas? Qual é o número de deslocamentos (i.e., cópias) de elementos efetuados?

Inicial:

1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

N.º de
comparações:

9

Final:

1									
---	--	--	--	--	--	--	--	--	--

N.º de Cópias:

36

Justifique a sua resposta:

Como a sequência inicial é composta por números idênticos, o algoritmo sempre que comparar 2 elementos vai ter que remover 1 deles e fazer deslocamentos, logo depois de 1 passagem o algoritmo elimina todos os elementos iguais.

- Indique uma sequência inicial com 10 elementos que conduza ao **pior caso do número de comparações** efetuadas. Qual é a sequência final obtida? Qual é o número de comparações efetuadas? Qual é o número de deslocamentos (i.e., cópias) de elementos efetuados?

Inicial:

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

N.º de
comparações:

45

Final:

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

N.º de Cópias:

0

Justifique a sua resposta:

Como a sequência inicial é composta por números diferentes entre si, o algoritmo nunca vai eliminar elementos repetidos, ou seja, vai ter de comparar todos os elementos entre eles.

- Determine formalmente a ordem de complexidade do algoritmo nas situações do **melhor caso** e do **pior caso**, considerando uma sequência de tamanho n . Tenha em atenção que deve obter expressões matemáticas exatas e simplificadas.

ANÁLISE FORMAL DO ALGORITMO - <u>NÚMERO DE COMPARAÇÕES</u>
MELHOR CASO - $B(N) = N-1$
PIOR CASO - $W(N) = (N^2-N)/2$

ANÁLISE FORMAL DO ALGORITMO - <u>NÚMERO DE DESLOCAMENTOS DE ELEMENTOS</u>
MELHOR CASO - $B(N) = 0$
PIOR CASO - $W(N) = (N^2-3N+2)/2$

APRESENTAÇÃO DO ALGORITMO
<pre> INT DESLOCAMENTOS = 0; INT COMPARACOES = 0; VOID REMOVE_DUPLICATES(INT* A, INT* N){ INT I, J, H; FOR (I = 0; I < *N; I++) { FOR (J = I+1; J < *N; J++) { COMPARACOES++; IF (A[I] == A[J]) { FOR (H = J; H < (*N)-1; H++) { A[H] = A[H+1]; DESLOCAMENTOS++; } J--; (*N)--; } } } } </pre>

```
}
```