

NOME: MIGUEL CABRAL

Nº MEC: 93091

## 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 SEGUINTE 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, 2, 3, 4, 5, 6, 7, 7, 8, 9}	0	1
{1, 2, 3, 4, 5, 5, 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 SEGUINTE QUESTÕES:

- QUAL É A SEQUÊNCIA (OU AS SEQUÊNCIAS) QUE CORRESPONDE(M) AO MELHOR CASO DO ALGORITMO?

A PRIMEIRA SEQUÊNCIA -> {1, 3, 4, 5, 5, 6, 7, 7, 8, 9}

- QUAL É A SEQUÊNCIA (OU AS SEQUÊNCIAS) QUE CORRESPONDE(M) AO PIOR CASO DO ALGORITMO?

A PENÚLTIMA -> {1, 2, 3, 4, 5, 6, 7, 8, 9, 9} E A ÚLTIMA -> {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

- DETERMINE O NÚMERO DE ADIÇÕES EFETUADAS NO CASO MÉDIO DO ALGORITMO (PARA  $N = 10$ ).

$$(1+2+3+4+5+6+7+8+9+9) / 10 = 5,4$$

- QUAL É A ORDEM DE COMPLEXIDADE DO ALGORITMO?

LINEAR  $\rightarrow O(N)$

- 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. FACA AS ANÁLISES NO VERSO DA FOLHA.

- CALCULE O VALOR DAS EXPRESSÕES PARA  $N = 10$  E COMPARE-OS COM OS RESULTADOS OBTIDOS EXPERIMENTALMENTE.

$B_c(10) = 1$ ,  $W_c(10) = 9$ ,  $A_c(10) = 5,4$ . QUE SÃO IGUAIS AOS RESULTADOS OBTIDOS EXPERIMENTALMENTE.

### APRESENTAÇÃO DO ALGORITMO

```
unsigned int nComp = 0;
int isArith(int a[], int n){
    assert(n>1);
    for(int i = 0; i < n-1; i++){
        nComp++;
        if(a[i+1]-a[i] != 1){
            return 0;
        }
    }
    return 1;
}
```

## ANÁLISE FORMAL DO ALGORITMO

MELHOR CASO -  $B(N) = 1 \rightarrow 1$  ITERAÇÃO DO CICLO  $\rightarrow O(1)$

$$\text{PIOR CASO - } W(N) = \sum_{i=1}^{n-1} 1 = n - 1 \rightarrow O(n - 1)$$

CASO MÉDIO -  $A(N) =$

$$\frac{1}{n} \left( \left( \sum_{i=1}^{n-1} i \right) + (n - 1) \right) = \frac{1}{n} \left( \frac{n*(n-1)}{2} + (n - 1) \right) = \frac{n^2+n-2}{2n}$$

**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.**

<u>SEQUÊNCIA INICIAL</u>	<u>SEQUÊNCIA FINAL</u>	<u>Nº DE COMPARAÇÕES</u>	<u>Nº DE DESLOCAMENTOS</u>
{1,2,2,2,3,3,4,5,8,8}	{1, 2, 3, 4, 5, 8 }	28	17
{1,2,2,2,3,3,3,3,8,8}	{1, 2, 3, 8 }	23	22
{1,2,3,2,1,3,4}	{1, 2, 3, 4 }	12	5
{1,2,5,4,7,0,3,9,6,8}	{1,2,5,4,7,0,3,9,6,8}	45	0
{1,1,1,1,1,1,1,1,1,1}	{1}	9	36

**DEPOIS DA EXECUÇÃO DO ALGORITMO RESPONDA ÀS SEGUINTE 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?**

[illegible]

Nº DE COMPARAÇÕES	Nº DE CÓPIAS
9	36

JUSTIFIQUE A SUA RESPOSTA:

COMO TODOS OS VALORES SÃO IGUAIS AO PRIMEIRO O ARRAY SÓ É PERCORRIDO UMA VEZ ELIMINANDO LOGO OS RESTANTES ELEMENTOS, FAZENDO ASSIM  $N-1$  COMPARAÇÕES.

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

Nº DE COMPARAÇÕES	Nº DE CÓPIAS
45	0

JUSTIFIQUE A SUA RESPOSTA:

SE OS ELEMENTOS FOREM TODOS DIFERENTES NENHUM É APAGADO DO ARRAY LOGO TERÁ DE SE FAZER COMPARAÇÕES ENTRE TODOS ( O 1º E OS RESTANTES, O 2º E OS RESTANTES ETC...) .

- 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. FAÇA AS ANÁLISES NO VERSO DA FOLHA.

---

APRESENTAÇÃO DO ALGORITMO

```
static int nComp= 0;
static int changes = 0;

void removeDuplicates(int* array, int* n){
    assert(*n>1);
    for(int i = 0; i < *n ; i++){
        for(int j = i + 1; j < *n; ){
            nComp++;
            if(array[j] == array[i]){
                for(int k = j ; k < *n - 1;k++){
                    array[k] = array[k + 1];
                    changes++;
                }
                (*n)--;
            }else{
                j++;
            }
        }
    }
}
```

## ANÁLISE FORMAL DO ALGORITMO

Nº DE COMPARAÇÕES

MELHOR CASO -  $B(N) = N-1 \rightarrow \text{PARA } N = 10 \text{ É } 9$ 
$$\text{PIOR CASO - } W(N) = \sum_{i=0}^{n-1} i = \frac{n*(n-1)}{2} \rightarrow \text{PARA } N = 10 \text{ É } 45$$

Nº DE DESLOCAMENTOS DE ELEMENTOS

MELHOR CASO -  $B(N) = 0 \rightarrow \text{PARA } N = 10 \text{ É } 0$

PIOR CASO -  $W(N) =$

$$\sum_{i=1}^{n-1} (i-1) = \sum_{i=1}^{n-1} (i) - \sum_{i=1}^{n-1} 1 = \frac{n(n-1)}{2} - (n-1) = \frac{n(n-1)}{2} - n + 1$$

$\rightarrow \text{PARA } N = 10 \text{ É } 36$