

AULA 4 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS

Seja uma dada sequência (*array*) de números inteiros ordenada por ordem não-decrescente. Pretende-se determinar se a sequência das diferenças entre os elementos sucessivos é uma sequência crescente contínua de inteiros. Por exemplo, a sequência {12, 15, 19, 24, 30, 37} define a sequência de diferenças {3, 4, 5, 6, 7}, que é uma sequência crescente contínua de inteiros a começar em 3. E a sequência {10, 14, 19, 25, 32, 40} define a sequência de diferenças {4, 5, 6, 7, 8}, que é uma sequência crescente contínua de inteiros a começar em 4. Mas a sequência {11, 15, 20, 26, 32, 40} define a sequência de diferenças {4, 5, 6, 6, ...} que não é uma sequência crescente contínua de inteiros, porque a quarta diferença devia ser 7 e é 6.

- Implemente uma função inteira **eficiente** e **eficaz** que, sem utilizar um *array* auxiliar, verifica se uma sequência com n elementos define uma sequência de diferenças que é crescente e contínua. Considere o resultado da função do tipo: 1 para verdadeiro e 0 para falso.

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

- Determine experimentalmente a **ordem de complexidade do número de operações aritméticas aditivas** (subtrações e adições) envolvendo os elementos da sequência e o cálculo do incremento da diferença. Considere as seguintes nove sequências de dez inteiros, todas diferentes e que cobrem todas as situações possíveis de execução do algoritmo. Calcule para cada uma delas se se verifica a condição pretendida e o número de operações aritméticas aditivas efetuadas.

1	3	5	7	9	11	20	25	27	29	Resultado		Nº de operações	
1	3	6	9	11	13	20	25	27	29	Resultado		Nº de operações	
1	3	6	10	11	13	20	25	27	29	Resultado		Nº de operações	
1	3	6	10	15	17	20	25	27	29	Resultado		Nº de operações	
1	3	6	10	15	21	22	25	27	29	Resultado		Nº de operações	
1	3	6	10	15	21	28	30	37	39	Resultado		Nº de operações	
1	3	6	10	15	21	28	36	39	49	Resultado		Nº de operações	
1	3	6	10	15	21	28	36	45	49	Resultado		Nº de operações	
1	3	6	10	15	21	28	36	45	55	Resultado		Nº de operações	

Depois da execução do algoritmo responda às seguintes questões:

- Qual é a sequência que corresponde ao melhor caso do algoritmo?
- Qual é a sequência (ou as sequências) que corresponde(m) ao pior caso do algoritmo? O que é que a(s) distingue?
- Determine o número de operações aditivas efetuadas no caso médio do algoritmo (para $N = 10$).
- Qual é a ordem de complexidade do algoritmo?
- Determine formalmente a ordem de complexidade do algoritmo nas situações do melhor caso, do pior caso e do caso médio, considerando uma sequência com dimensão N (com $N > 2$). Tenha em atenção que deve obter expressões matemáticas exatas e simplificadas.

Faça as análises no verso da folha.

APRESENTAÇÃO DO ALGORITMO**ANÁLISE FORMAL DO ALGORITMO**

MELHOR CASO - $B(N)$ =

PIOR CASO - $W(N)$ =

CASO MÉDIO - $A(N)$ =

- Calcule o valor de cada uma das expressões para $N = 10$ e compare-os com os resultados obtidos experimentalmente.

NOME:

Nº MEC:

Seja uma dada sequência (*array*) aleatória de números inteiros. Pretende-se determinar a maior repetição de elementos que exista na sequência. Por exemplo, na sequência { 1, 8, 2, 3, 3, 4, 2, 5, 1, 8 } existem vários números (como sejam, o 1, o 2, o 3 e o 8) que aparecem repetidos duas vezes cada. Na sequência { 1, 1, 2, 2, 2, 1, 4, 4, 5, 2 } o número 2 aparece repetido quatro vezes. E na sequência { 2, 2, 2, 3, 3, 1, 3, 2, 3, 3 } o número 3 aparece repetido cinco vezes. O algoritmo deve ser suficientemente eficiente para que, assim que não houver a possibilidade de existir um número repetido mais vezes do que a maior repetição anteriormente detetada a pesquisa deve parar. Por exemplo, a sequência { 5, 1, 1, 1, 8, 5, 4, 5, 8, 1 } em que o elemento 1 aparece repetido quatro vezes impede que a partir do índice 6 do *array*, quando restam apenas 4 elementos na sequência, exista algum elemento que esteja repetido mais do que quatro vezes.

- Implemente uma função inteira **eficiente** e **eficaz** que determina a maior repetição de um número que existe numa sequência com n elementos. A função devolve o número de vezes que o número mais repetido aparece na sequência. Depois de validar o algoritmo apresente-o no verso da folha
- Determine experimentalmente o **número de operações de comparação** envolvendo os elementos da sequência e a determinação da maior repetição. Considere as seguintes onze sequências de dez inteiros, todas diferentes e que cobrem todas as situações possíveis de execução do algoritmo. Calcule para cada uma delas a maior repetição e o número de comparações efetuadas.

1	1	1	1	1	1	1	1	1	1	Resultado		Nº de operações	
1	1	1	1	1	3	1	1	1	1	Resultado		Nº de operações	
1	1	1	4	1	1	4	1	1	1	Resultado		Nº de operações	
1	1	1	1	1	1	2	5	1	9	Resultado		Nº de operações	
1	1	1	1	9	1	2	5	1	9	Resultado		Nº de operações	
1	1	1	9	1	2	8	3	7	1	Resultado		Nº de operações	
1	1	1	9	5	2	8	1	9	9	Resultado		Nº de operações	
1	1	3	9	5	2	8	7	9	9	Resultado		Nº de operações	
1	1	6	0	5	2	8	7	9	9	Resultado		Nº de operações	
1	4	6	0	5	2	8	7	9	9	Resultado		Nº de operações	
1	3	6	0	5	2	8	7	11	9	Resultado		Nº de operações	

Depois da execução do algoritmo responda às seguintes questões:

- Quais são as sequências que correspondem ao melhor caso do algoritmo? O que é que as distingue?
- Quais são as sequências que correspondem ao pior caso do algoritmo? O que é que as distingue?
- Determine o número de comparações efetuadas no caso médio do algoritmo (para $N = 10$).
- Qual é a ordem de complexidade do algoritmo?
- Determine formalmente a ordem de complexidade do algoritmo, nas situações do melhor caso, do pior caso e do caso médio, considerando uma sequência com dimensão N (com $N > 2$). Tenha em atenção que deve obter expressões matemáticas exatas e simplificadas.

Faça as análises no verso da folha.

APRESENTAÇÃO DO ALGORITMO**ANÁLISE FORMAL DO ALGORITMO**

MELHOR CASO - $B(N)$ =

PIOR CASO - $W(N)$ =

CASO MÉDIO - $A(N)$ =

- Calcule o valor de cada uma das expressões para $N = 10$ e compare-os com os resultados obtidos experimentalmente.

NOME:

Nº MEC: