Aula 6 - Análise da Complexidade de Algoritmos Recursivos

**\*\*\* Entregue, num ficheiro ZIP, este guião preenchido e o código desenvolvido \*\*\***

Implemente os seguintes **algoritmos recursivos** – **sem recorrer a funções de arredondamento** (floor e ceil) – e analise o **número de chamadas recursivas** executadas por cada algoritmo.

Deve utilizar **aritmética inteira**: n/4 é igual a e (n+3)/4 é igual a .

* **Preencha a tabela da página seguinte** com o resultado de cada função e o número de chamadas recursivas para os sucessivos valores de n.
* Analisando os dados da tabela, estabeleça uma ordem de complexidade para cada algoritmo.

|  |
| --- |
| T2(n) = O  T3(n) = O(n) |

* Escreva uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função **.** Obtenha, depois, uma **expressão exata e simplificada;** determine a sua **ordem de complexidade**. Compare a expressão obtida com os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico**.

|  |
| --- |
| é de notar que aqui o valor é 2 e não 1 porque é contada a chamada da função T1 pela função main  aproxima-se de  , pois  ∈ O() |

**­­**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **T1(n)** | **Nº de Chamadas Recursivas** | **T2(n)** | **Nº de Chamadas Recursivas** | **T3(n)** | **Nº de Chamadas Recursivas** |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 1 | 2 | 1 |
| 3 | 3 | 2 | 3 | 1 | 3 | 1 |
| 4 | 5 | 3 | 6 | 3 | 6 | 2 |
| 5 | 6 | 3 | 8 | 3 | 8 | 3 |
| 6 | 7 | 3 | 9 | 3 | 9 | 3 |
| 7 | 8 | 3 | 10 | 3 | 10 | 3 |
| 8 | 10 | 3 | 12 | 3 | 12 | 2 |
| 9 | 11 | 3 | 14 | 3 | 14 | 3 |
| 10 | 12 | 3 | 15 | 3 | 15 | 3 |
| 11 | 13 | 3 | 16 | 3 | 16 | 3 |
| 12 | 15 | 3 | 18 | 3 | 18 | 2 |
| 13 | 16 | 3 | 22 | 5 | 22 | 4 |
| 14 | 17 | 3 | 23 | 5 | 23 | 4 |
| 15 | 18 | 3 | 24 | 5 | 24 | 4 |
| 16 | 21 | 4 | 28 | 7 | 28 | 3 |
| 17 | 22 | 4 | 31 | 7 | 31 | 6 |
| 18 | 23 | 4 | 32 | 7 | 32 | 6 |
| 19 | 24 | 4 | 33 | 7 | 33 | 6 |
| 20 | 26 | 4 | 36 | 7 | 36 | 4 |
| 21 | 27 | 4 | 38 | 7 | 38 | 7 |
| 22 | 28 | 4 | 39 | 7 | 39 | 7 |
| 23 | 29 | 4 | 40 | 7 | 40 | 7 |
| 24 | 31 | 4 | 42 | 7 | 42 | 4 |
| 25 | 32 | 4 | 44 | 7 | 44 | 7 |
| 26 | 33 | 4 | 45 | 7 | 45 | 7 |
| 27 | 34 | 4 | 46 | 7 | 46 | 7 |
| 28 | 36 | 4 | 48 | 7 | 48 | 4 |
| 29 | 37 | 4 | 51 | 7 | 51 | 6 |
| 30 | 38 | 4 | 52 | 7 | 52 | 6 |
| 31 | 39 | 4 | 53 | 7 | 53 | 6 |
| 32 | 42 | 4 | 56 | 7 | 56 | 3 |
| 33 | 43 | 4 | 59 | 7 | 59 | 6 |
| 34 | 44 | 4 | 60 | 7 | 60 | 6 |
| 35 | 45 | 4 | 61 | 7 | 61 | 6 |
| 36 | 47 | 4 | 64 | 7 | 64 | 4 |
| 37 | 48 | 4 | 66 | 7 | 66 | 7 |
| 38 | 49 | 4 | 67 | 7 | 67 | 7 |
| 39 | 50 | 4 | 68 | 7 | 68 | 7 |
| 40 | 52 | 4 | 70 | 7 | 70 | 4 |
| 41 | 53 | 4 | 72 | 7 | 72 | 7 |
| 42 | 54 | 4 | 73 | 7 | 73 | 7 |
| 43 | 55 | 4 | 74 | 7 | 74 | 7 |
| 44 | 57 | 4 | 76 | 7 | 76 | 4 |
| 45 | 58 | 4 | 79 | 7 | 79 | 6 |
| 46 | 59 | 4 | 80 | 7 | 80 | 6 |
| 47 | 60 | 4 | 81 | 7 | 81 | 6 |
| 48 | 63 | 4 | 84 | 7 | 84 | 3 |
| 49 | 64 | 4 | 89 | 9 | 89 | 7 |
| 50 | 65 | 4 | 90 | 9 | 90 | 7 |
| 51 | 66 | 4 | 91 | 9 | 91 | 7 |
| 52 | 68 | 4 | 96 | 11 | 96 | 5 |
| 53 | 69 | 4 | 98 | 11 | 98 | 9 |
| 54 | 70 | 4 | 99 | 11 | 99 | 9 |
| 55 | 71 | 4 | 100 | 11 | 100 | 9 |
| 56 | 73 | 4 | 102 | 11 | 102 | 5 |
| 57 | 74 | 4 | 104 | 11 | 104 | 9 |
| 58 | 75 | 4 | 105 | 11 | 105 | 9 |
| 59 | 76 | 4 | 106 | 11 | 106 | 9 |
| 60 | 78 | 4 | 108 | 11 | 108 | 5 |
| 61 | 79 | 4 | 113 | 13 | 113 | 8 |
| 62 | 80 | 4 | 114 | 13 | 114 | 8 |
| 63 | 81 | 4 | 115 | 13 | 115 | 8 |
| 64 | 85 | 5 | 120 | 15 | 120 | 4 |

* Escreva uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função **. Considere o caso particular e** obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

|  |
| --- |
| Pelo desenvolvimento telescópico:  aproxima-se de  , daqui podemos dizer que  Pelo Teorema Mestre:  C(n) = 2 \* (n/4) + 1 daqui retiramos que a = 2, b = 4, f(n) = 1  f(n) ∈ O(1) ( f(n) ∈ O( ( d = 0  a ? ( a > 1, quando a > b a complexidade é dada por  e retiramos que O = O |

* Pode **generalizar a ordem de complexidade** que acabou de obter para todo o n? **Justifique.**

|  |
| --- |
| Pela smoothness rule se a ordem de complexidade do teorema mestre for de ordem logarítmica, linear ou polinomial então os casos particulares podem ser generalizados. Como a complexidade é O() então podemos generalizar a ordem de complexidade para todos os casos. |

* Obtenha uma **expressão recorrente** para o **número de chamadas recursivas** efetuadas pela função

|  |
| --- |
| Assumindo que R(n) é a função que expressa as chamadas recursivas da função T3(n) |

* **Considere o caso particular e** obtenha uma **expressão exata e simplificada;** determine a **ordem de complexidade** para esse caso particular. Compare a expressão obtida com os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

|  |
| --- |
| Pelo Desenvolvimento Telescópico:  Para n = apenas entramos o segundo ramo da função  Logo R(n) =  Pelo Teorema Mestre:  daqui retiramos que a = 2, b = 4 e f(n) = n  a ? ( 2 < 4, quando a < b a complexidade é dada por  Daqui retiramos que a complexidade é O(n) |

* Pode **generalizar a ordem de complexidade** que acabou de obter para todo o n? **Justifique.**

|  |
| --- |
| Não porque os valores de N entre casos particulares não estão dentro dessa complexidade, ou seja, para a ordem poder ser generalizada os valores de R(n) quando a < n < b, sendo a e b casos particulares de T3 não obedecem a R(a) < R(n) < R(b).  Resumidamente a complexidade para múltiplos de 4 é diferente para a complexidade de números que não são. |

* Atendendo às **semelhanças entre e**  estabeleça uma **ordem de complexidade para . Justifique.**

|  |
| --- |
| Vistos que a diferença de T3 para T2 é que T3 tem um caso particular para o qual faz menos chamadas recursivas do que o caso geral para T2 e vistos também que o caso geral de T3 é igual ao caso geral de T2 podemos dizer que a complexidade de T3 está sempre dentro da ordem de complexidade de T2, isto porque T3 tem sempre o mesmo número de chamadas recursivas ou menos. |