|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | **Gil Lopes Teixeira** | **N.º Mec:** | **88194** |

Aula 5 - 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/3 é igual a e (n+2)/3 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?

|  |
| --- |
| Olhando para a tabela t1(0) = 0 e O(t1(n)) = 1 + ;  t2(0) = t2(1) = t2(2) = 0 e O(t2(n)) é quanto muito linear;  t3(0) = t3(1) = t3(2) = 0 e O(t3(n)) é também quanto muito linear embora que com menos chamadas. |

* 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 a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico**.

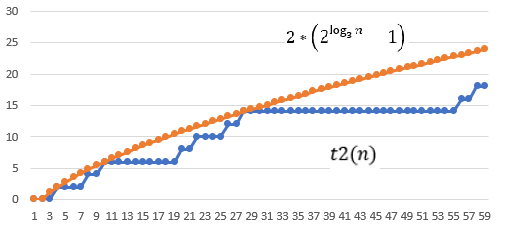
|  |
| --- |
| Expressão recorrente =  Expressão exata =  para k = ,  c1(n) = ;  O que está de acordo com os resultados experimentais obtidos.  Complexidade = O(); |

**­­**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | T1(n) | Nº de Chamadas Recursivas | T2(n) | Nº de Chamadas Recursivas | T3(n) | Nº de Chamadas Recursivas |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 2 | 2 | 1 | 2 | 0 | 2 | 0 |
| 3 | 4 | 2 | 5 | 2 | 5 | 1 |
| 4 | 5 | 2 | 7 | 2 | 7 | 2 |
| 5 | 6 | 2 | 8 | 2 | 8 | 2 |
| 6 | 8 | 2 | 10 | 2 | 10 | 1 |
| 7 | 9 | 2 | 14 | 4 | 14 | 3 |
| 8 | 10 | 2 | 15 | 4 | 15 | 3 |
| 9 | 13 | 3 | 19 | 6 | 19 | 2 |
| 10 | 14 | 3 | 22 | 6 | 22 | 5 |
| 11 | 15 | 3 | 23 | 6 | 23 | 5 |
| 12 | 17 | 3 | 26 | 6 | 26 | 3 |
| 13 | 18 | 3 | 28 | 6 | 28 | 6 |
| 14 | 19 | 3 | 29 | 6 | 29 | 6 |
| 15 | 21 | 3 | 31 | 6 | 31 | 3 |
| 16 | 22 | 3 | 34 | 6 | 34 | 5 |
| 17 | 23 | 3 | 35 | 6 | 35 | 5 |
| 18 | 26 | 3 | 38 | 6 | 38 | 2 |
| 19 | 27 | 3 | 43 | 8 | 43 | 6 |
| 20 | 28 | 3 | 44 | 8 | 44 | 6 |
| 21 | 30 | 3 | 49 | 10 | 49 | 4 |
| 22 | 31 | 3 | 51 | 10 | 51 | 8 |
| 23 | 32 | 3 | 52 | 10 | 52 | 8 |
| 24 | 34 | 3 | 54 | 10 | 54 | 4 |
| 25 | 35 | 3 | 59 | 12 | 59 | 7 |
| 26 | 36 | 3 | 60 | 12 | 60 | 7 |
| 27 | 40 | 4 | 65 | 14 | 65 | 3 |
| 28 | 41 | 4 | 69 | 14 | 69 | 9 |

* 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 a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

|  |
| --- |
| Expressão recorrente =  Expressão exata para =  para k =,  c2(n) = ;  Teorema mestre: Como a função de recorrência é definida por ,  com a = 2, b=3, d=0, , logo c2(n) o que está de acordo com os resultados experimentais. |



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

|  |
| --- |
| Como a c2(n) é não decrescente, pela Regra da suavidade podemos generalizar a ordem de complexidade para todo o n. |

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

|  |
| --- |
| Expressão recorrente = |

* **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 a os dados da **tabela**. Sugestão: use o **desenvolvimento telescópico** e confirme o resultado obtido usando o **Teorema Mestre**.

|  |
| --- |
| Expressão exata para :  para k =    Teorema mestre: Como a função de recorrência é definida por ,  com a = 1, b=3, c = 0, d=0, , logo c3(n) o que está de acordo com os resultados experimentais. |

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

|  |
| --- |
| A ordem de complexidade calculada foi para o caso específico de. Quando isto não é verdade a função t3(n) usa uma expressão diferente para o cáculo do resultado pelo que não é possível generalizar a ordem de complexidade. |

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

|  |
| --- |
| A ordem de complexidade de t3(n) para é inferior à de t2(n). Nos casos em que o comportamento de t3(n) é semelhante ao de t2(n) pelo que podemos dizer que no caso limite |

