

**AULA 6 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS RECURSIVOS (NÚMEROS DE MOTZKIN)****\*\*\* Entregue, num ficheiro ZIP, este guião preenchido e o código desenvolvido \*\*\***

- Os números de Motzkin 1, 1, 2, 4, 9, 21, 51,... (<https://oeis.org/A001006>) são definidos pela seguinte relação de recorrência:

$$\text{Motzkin}(n) = \begin{cases} 1, & \text{se } n = 0 \text{ e } n = 1 \\ \text{Motzkin}(n-1) + \sum_{k=0}^{n-2} \text{Motzkin}(k) \times \text{Motzkin}(n-2-k), & \text{se } n > 1 \end{cases}$$

**Função Recursiva**

- Implemente uma **função recursiva Motzkin(n)** que use diretamente a relação de recorrência acima, **sem qualquer simplificação**.
- Construa um programa para executar a função **Motzkin(n)** para **sucessivos valores de n** e que permita **contar o número total de multiplicações efetuadas** para cada valor de n.
- Preencha a as primeiras colunas tabela seguinte** com o resultado da função recursiva e o número de multiplicações efetuadas para os sucessivos valores de n.

n	Motzkin(n) – Versão Recursiva	Nº de Multiplicações	Motzkin(n) – Versão de Programação Dinâmica	Nº de Multiplicações
0	1	0	1	0
1	1	0	1	0
2	2	1	2	1
3	4	3	4	3
4	9	8	9	6
5	21	20	21	10
6	51	49	51	15
7	127	119	127	21
8	323	288	323	28
9	835	696	835	36
10	2188	1681	2188	45
11	5798	4059	5798	55
12	15511	9800	15511	66
13	41835	23660	41835	78
14	113634	57121	113634	91
15	310572	137903	310572	105

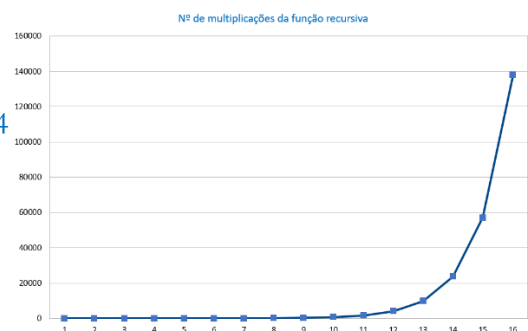
- Analisando os dados da tabela, estabeleça uma **ordem de complexidade** para a **função recursiva**.

Analisando os dados da tabela:

$$\rightarrow \frac{\text{número de multiplicações}(i)}{\text{número de multiplicações}(i-1)} \approx 2,414;$$

O número de multiplicações é obtido por multiplicar 2,414 sucessivamente.

Logo, a ordem de complexidade é exponencial,  $O(2,414^n)$ .

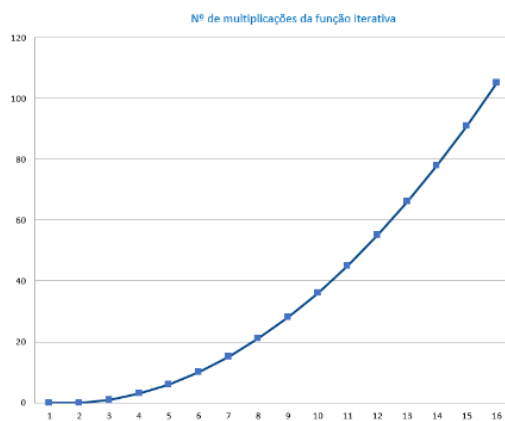


### Programação Dinâmica

- Uma forma alternativa de resolver alguns problemas recursivos, para evitar o cálculo repetido de valores, consiste em efetuar esse cálculo de baixo para cima (*"bottom-up"*), ou seja, de **Motzkin(0)** para **Motzkin(n)**, e utilizar um *array* para manter os valores entretanto calculados. Este método designa-se por **programação dinâmica** e reduz o tempo de cálculo à custa da utilização de mais memória para armazenar os valores intermédios.
- Usando **programação dinâmica**, implemente uma **função iterativa** para calcular Motzkin(n). **Não utilize um array global.**
- Construa um programa para executar a função iterativa que desenvolveu para **sucessivos valores de n** e que permita **contar o número de multiplicações efetuadas** para cada valor de n.
- **Preencha as últimas colunas tabela anterior** com o resultado da função iterativa e o número de multiplicações efetuadas para os sucessivos valores de n.
- Analisando os dados da tabela, estabeleça uma **ordem de complexidade** para a **função iterativa**.

Analisando os dados da tabela a ordem de complexidade é quadrática  $O(n^2)$ .

A partir do gráfico, que relaciona número de multiplicações efetuadas para os sucessivos valores de n, observamos que o comportamento geral do algoritmo é descrito por uma função quadrática.



## Função Recursiva – Análise Formal da Complexidade

- Escreva uma **expressão recorrente** (direta) para o **número de multiplicações** efetuadas pela função recursiva Motzkin(n). Obtenha, depois, uma **expressão recorrente simplificada**. Note que  $\sum_{k=0}^{n-2} \text{Mult}(k) = \sum_{k=0}^{n-2} \text{Mult}(n-2-k)$ . **Sugestão:** efetue a subtração  $\text{Mult}(n) - \text{Mult}(n-1)$ .

$$\begin{aligned}
 \text{mult}(n) &= \begin{cases} 0 & n=0, n=1 \\ \text{mult}(n-1) + \sum_{k=0}^{n-2} (\text{mult}(k) + \text{mult}(n-2-k) + 1) & n \geq 2 \end{cases} \\
 \text{mult}(n-1) + \sum_{k=0}^{n-2} (\text{mult}(k) + \text{mult}(n-2-k) + 1) &= \\
 = \text{mult}(n-1) + \sum_{k=0}^{n-2} \text{mult}(k) + \sum_{k=0}^{n-2} (\text{mult}(n-2-k)) + \sum_{k=0}^{n-2} 1 &= \\
 = \text{mult}(n-1) + 2 \sum_{k=0}^{n-2} \text{mult}(k) + (n-1) & \\
 \text{mult}(n) = \text{mult}(n-1) + 2 \sum_{k=0}^{n-2} \text{mult}(k) + (n-1) & \\
 = \text{mult}(n-1) + 2 \sum_{k=0}^{n-3} \text{mult}(k) + (n-1) + 2 \text{mult}(n-2) & \\
 \text{mult}(n-1) = \text{mult}(n-2) + 2 \sum_{k=0}^{n-3} \text{mult}(k) + (n-2) & \\
 \text{mult}(n) - \text{mult}(n-1) = \text{mult}(n-1) + (n-1) + 2 \text{mult}(n-2) + 2 \sum_{k=0}^{n-3} \text{mult}(k) & \\
 - (\text{mult}(n-2) + 2 \sum_{k=0}^{n-3} \text{mult}(k) + (n-2)) &= \\
 = \text{mult}(n-1) + \text{mult}(n-2) + 1 & \\
 \text{mult}(n) - \text{mult}(n-1) = \text{mult}(n-1) + \text{mult}(n-2) + 1 \quad (\Leftrightarrow) & \\
 \Leftrightarrow \text{mult}(n) = 2 \text{mult}(n-1) + \text{mult}(n-2) + 1 &
 \end{aligned}$$

- A equação de recorrência obtida é uma **equação de recorrência linear não homogénea**. Considere a correspondente **equação de recorrência linear homogénea**. Determine as raízes do seu **polinómio característico**. Sem determinar as constantes associadas, escreva a **solução da equação de recorrência linear não homogénea**.

$$\begin{aligned}
 \text{mult}(n) &\leq 2 \text{mult}(n-1) + \text{mult}(n-2) \\
 \text{mult}(n) &= 2 \text{mult}(n-1) + \text{mult}(n-2) \\
 \text{equação característica: } x^2 - 2x - 1 &= 0 \\
 x = \frac{2 \pm \sqrt{(-2)^2 - 4 \cdot 1 \cdot (-1)}}{2 \cdot 1} \quad (\Leftrightarrow) \quad x = 1 + \sqrt{2} \quad \vee \quad x = 1 - \sqrt{2} \quad (\Leftrightarrow) \\
 &\quad \Leftrightarrow \quad x \approx 2,414 \quad \vee \quad x \approx -0,414 \\
 \text{mult}(n) &= 2,414^n \cdot C_1 + (-0,414)^n \cdot C_2 \quad C_1, C_2 \in \mathbb{R} \\
 \frac{\text{número de multiplicações}(i)}{\text{número de multiplicações}(i-1)} &\approx 2,414 \quad (1 + \sqrt{2}) \\
 \textcircled{1} (2,414^n) &, \text{ ordem de complexidade exponencial}
 \end{aligned}$$

- Usando a solução da equação de recorrência obtida acima, determine a **ordem de complexidade do número de multiplicações** efetuadas pela função recursiva. **Compare** a ordem de complexidade que acabou de obter com o resultado da **análise experimental**.

Da equação acima obtemos a seguinte expressão:

$$\rightarrow \text{mult}(n) = 2,414^n * C_1 + (-0,414)^n * C_2, \text{ tal que } C_1, C_2 \in \mathbf{R};$$

Da tabela retiramos que:

$$\rightarrow \frac{\text{número de multiplicações}(i)}{\text{número de multiplicações}(i-1)} \approx 2,414;$$

Concluimos que, pela equação de recorrência e experimentalmente, a ordem de complexidade do número de multiplicações é exponencial,  $O(2,414^n)$ .

#### Programação Dinâmica – Análise Formal da Complexidade

- Considerando o número de multiplicações efetuadas pela função iterativa, efetue a análise formal da sua complexidade. Obtenha uma **expressão exata e simplificada para o número de multiplicações** efetuadas.

$$\text{mult}(n) = \sum_{k=2}^n \sum_{m=0}^{k-2} 1 = \sum_{k=2}^n (k-1) = \sum_{k=2}^n k - \sum_{k=2}^n 1 = \frac{(n-1)(n+2)}{2} - (n-1) = \frac{n^2-n}{2}$$

- Usando a expressão obtida acima, determine a **ordem de complexidade do número de multiplicações** efetuadas pela função iterativa. **Compare** a ordem de complexidade que acabou de obter com o resultado da **análise experimental**.

A partir da expressão calculada,  $\frac{n^2-n}{2}$ , a ordem de complexidade do número de multiplicações é quadrática,  $O(n^2)$ , assim como foi concluído experimentalmente.