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:

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

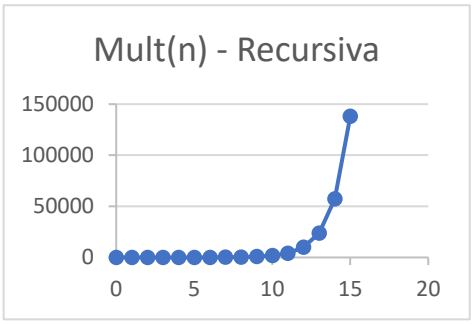
Função Recursiva

- 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**.

Ao analisar o valor do número de multiplicações em função de n da função recursiva, à primeira vista, pomos a hipótese de a ordem de complexidade ser exponencial (também visível na curva apresentada no gráfico).



Logo para tentar verificar se a complexidade é de facto exponencial vamos efetuar as divisões $\frac{Mult(n+1)}{Mult(n)}$ para sucessivos valores de n e verificar se os valores obtidos convergem para uma certa constante.

n	Mult(n) - Recursiva	Mult(n+1)/Mult(n)
0	0	#DIV/0!
1	0	#DIV/0!
2	1	3
3	3	2,66666667
4	8	2,5
5	20	2,45
6	49	2,42857143
7	119	2,42016807
8	288	2,41666667
9	696	2,41522989
10	1681	2,41463415
11	4059	2,41438778
12	9800	2,41428571
13	23660	2,41424345
14	57121	2,41422594

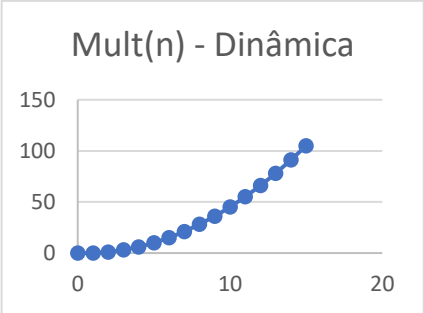
Após análise dos resultados concluímos que à medida que o valor de n aumenta, o resultado de $\frac{Mult(n+1)}{Mult(n)}$ converge para aproximadamente 2,41. Consequentemente podemos afirmar que a ordem de complexidade da função recursiva é exponencial, $O(a^n)$, em que a toma o valor 2,41.

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**.

Seguindo uma metodologia semelhante à recursiva, ao analisar os valores da tabela ponderamos a hipótese de a complexidade ser quadrática (visível também na curva do gráfico).



Logo assim sendo vamos efetuar as divisões $\frac{\log\left(\frac{Mult(n+1)}{Mult(n)}\right)}{\log\left(\frac{n+1}{n}\right)}$, para os sucessivos valores de n, e verificar se o resultado converge para um certo valor.

n	Mult(n) - Dinâmica	log(Mult(n+1)/Mult(n))/ log((n+1)/n)
0	0	#DIV/0!
1	0	#DIV/0!
2	1	2,70951129
3	3	2,40942084
4	6	2,28922423
5	10	2,22390109
6	15	2,18274896
7	21	2,15441528
8	28	2,1337065
9	36	2,11790489
10	45	2,10544871
11	55	2,09537607
12	66	2,08706189
13	78	2,08008228
14	91	2,08008228

Analisando os resultados obtidos concluímos que as divisões efetuadas convergem para o valor 2. Logo podemos afirmar que a complexidade é do tipo $O(n^a)$, em que $a = 2$, ou seja complexidade 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 $\mathbf{Mult(n) - Mult(n-1)}$.

A partir da observação da relação de recorrência Motzkin(n), podemos obter uma expressão recorrente direta para o número de multiplicações:

$$Mult(n) = \begin{cases} 0\,, n \leq 1 \\ Mult(n-1) + \sum_{k=0}^{n-2} (Mult(k) + Mult(n-2-k) + 1)\ , n \geq 2 \end{cases}$$

Expressão que neste caso pode ser simplificada tendo em conta que $Mult(0) = Mult(1) = 0$:

$$Mult(n) = Mult(n-1) + \sum_{k=0}^{n-2} (2 \cdot Mult(k) + 1) \ = \ Mult(n-1) + 2 \cdot \sum_{k=0}^{n-2} \big(Mult(k)\big) + (n-1) \quad , n \geq 2$$

Vamos agora calcular a expressão recorrente simplificada, com auxílio da fórmula direta obtida anteriormente, através da seguinte subtração:

$$\begin{aligned} Mult(n) - Mult(n-1) &= \left(Mult(n-1) + 2 \cdot \sum_{k=0}^{n-2} \big(Mult(k)\big) + (n-1) \right) - \left(Mult(n-2) + 2 \cdot \sum_{k=0}^{n-3} \big(Mult(k)\big) + (n-2) \right) \\ &= Mult(n-1) + 2 \cdot Mult(n-2) + 2 \cdot \sum_{k=0}^{n-3} \big(Mult(k)\big) + n - 1 - Mult(n-2) - 2 \cdot \sum_{k=0}^{n-3} \big(Mult(k)\big) - n + 2 \\ &= Mult(n-1) + Mult(n-2) + 1 \end{aligned}$$

E assim sendo através do resultado obtido anteriormente, concluímos que a expressão recorrente simplificada é:

$$Mult(n) = 2 \cdot Mult(n-1) + Mult(n-2) + 1 \qquad , n \geq 2$$

- 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**.

Equação de recorrência:

$$Mult(n) = 2 \cdot Mult(n-1) + Mult(n-2) + 1 \Leftrightarrow Mult(n) - 2 \cdot Mult(n-1) - Mult(n-2) = 1$$

O que representa uma equação de recorrência linear não homogénea, cuja solução é do tipo

$$Mult(n) = Mult^1(n) + Mult^2(n)$$

Cálculo do $Mult^1(n)$

(1) Cálculo das raízes do polinómio característico da equação de recorrência linear homogénea associada
Equação de recorrência linear homogénea: $Mult(n) - 2 \cdot Mult(n-1) - Mult(n-2) = 0$

$$x^2 - 2x - 1 = 0 \Leftrightarrow x = \frac{-(-2) \pm \sqrt{(-2)^2 - 4 \cdot 1 \cdot (-1)}}{2 \cdot 1} \Leftrightarrow x = \frac{2 \pm \sqrt{8}}{2} \Leftrightarrow x = 1 \pm \sqrt{2}$$

(2) Conclusão
Concluímos assim que $Mult^1(n) = (1+\sqrt{2})^n A + (1-\sqrt{2})^n B$ em que A e B são as constantes associadas.

Cálculo do $Mult^2(n)$

(1) $Mult^2(n)$ é calculado tendo em conta que $f(n) = 1$ (ver equação de recorrência)
$Mult^2(n) = Cn^r$ em que C é uma constante e r é a multiplicidade de 1 enquanto raiz característica da equação linear homogénea obtida anteriormente ($r = 0$, neste caso).

(2) Conclusão
Concluímos que $Mult^2(n) = Cn^0 = C$, em que C é uma constante associada.

Podemos assim afirmar que a solução da equação de recorrência linear não homogénea é:
$Mult(n) = (1+\sqrt{2})^n A + (1-\sqrt{2})^n B$ + C

- 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**

Analisando a solução obtida anteriormente $Mult(n) = (1+\sqrt{2})^n A + (1-\sqrt{2})^n B$ + C, em que A,B e C são constantes, podemos determinar a ordem de complexidade do número de multiplicações efetuadas pela função recursiva.

Ao saber que para a determinação da ordem de complexidade devemos desprezar constantes e termos de menor ordem, chegamos então à conclusão de que $(1+\sqrt{2})^n$ é o termo de maior ordem, logo todos os outros podem ser desprezados

Assim sendo concluímos que a ordem de complexidade do algoritmo é $O\left((1+\sqrt{2})^n\right)$, que neste caso representa uma complexidade exponencial.

A partir da análise experimental concluímos que a complexidade era $O(a^n)$ em que $a \cong 2{,}41$. Comparando o resultado obtido anteriormente, podemos verificar que $1+\sqrt{2} \cong 2{,}41$ o que nos indica que o valor da complexidade é o mesmo que o obtido na análise experimental.

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.

Considerando as condições iniciais, $Mult(0) = Mult(1) = 0$, temos que:

$$Mult(n) = \sum_{i=2}^{n}\left(\sum_{k=0}^{i-2} 1\right) = \sum_{i=2}^{n} i - 1 = \frac{1+n-1}{2}\cdot(n-1) = \frac{n}{2}\cdot(n-1) = \frac{n^2-n}{2}$$

Ao analisar o resultado obtido chegamos à conclusão de que o número de multiplicações efetuadas pela função iterativa, tem uma complexidade quadrática, $O(n^2)$.

- 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 expressão obtida anteriormente para o número de multiplicações foi $Mult(n) = \frac{n^2-n}{2}$

Assim sendo facilmente concluímos que a ordem de complexidade para o número de multiplicações efetuadas pela versão iterativa é quadrática, $O(n^2)$, visto que o termo de maior ordem neste caso é $\frac{n^2}{2}$.

O valor obtido através da análise experimental foi também de complexidade quadrática $O(n^2)$.