

**TD2****EXERCICE 1 :**

Soit la suite  $U_n$  définie par :

$$\begin{cases} U_n = U_{n-1} \times U_{n-2} + U_{n-3} \\ U_0 = 1 \\ U_1 = 1 \\ U_2 = 1 \end{cases}$$

**Question :**

- Donner un algorithme récursif qui calcule  $U_n$
- Évaluer sa complexité.

**EXERCICE 2 : TRIANGLE DE PASCAL**

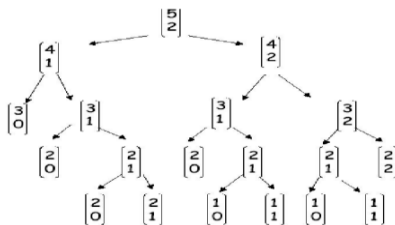
On veut calculer les coefficients binomiaux  $C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}$ . Rappelons les propriétés suivantes :

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \text{ pour } 0 < k < n$$

$$\binom{n}{n} = 1 \text{ et } \binom{n}{0} = 1$$

**Question :**

- Donner un algorithme récursif qui calcule  $\binom{n}{k}$
- Évaluer sa complexité.



	0	1	2	3	...	n-1	n
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
⋮	⋮	⋮	⋮	⋮	⋮		
n-1	1	n-1	$\binom{n-1}{2}$	$\binom{n-1}{3}$	...	1	
n	1	n	$\binom{n}{2}$	$\binom{n}{3}$	...	n	1

**EXERCICE 3 :**

- Écrire une fonction qui permet de calculer la somme des éléments d'une matrice carrée
- Évaluer sa complexité.

**EXERCICE 4 :**

1. Écrire une fonction itérative puissanceIterative (a, n) qui permet de calculer  $a^n$ . Rq. En utilisant seulement les opérateurs simples (+, -, \*, /)
2. Évaluer sa complexité.
3. Écrire une fonction récursive puissanceRecursive (a, n) qui permet de calculer  $a^n$ .
4. Évaluer sa complexité.

**EXERCICE 5 :**

Les nombres de Fibonacci sont définis par la récurrence :

- $F_0 = 1$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$  pour  $n \geq 2$

1. Ecrire une fonction récursive permettant de calculer Fib (n)
2. Évaluer sa complexité.
3. Ecrire une fonction itérative permettant de calculer Fib (n)
4. Évaluer sa complexité.