# Exercices : Exponentiation.

**Exercice 1.** ***Une application de l'exponentiation « rapide »***

On appelle **matrice « carrée »** d'ordre 2 un tableau constitué de 4 nombres réels :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad (a,b,c,d) \in \mathbb{R}^4$$

Pour la suite de l'exercice, on désignera plus simplement par « matrice » une matrice carrée d'ordre 2.

Il est possible d'additionner (ou soustraire) des matrices, mais dans cet exercice, nous allons utiliser la multiplication. Celle-ci est définie comme suit :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} = \begin{pmatrix} aa'+bc' & ab'+bd' \\ a'c+c'd & b'c+dd' \end{pmatrix}$$

*(On pourra aussi donner une « disposition graphique » du produit matriciel)*

Comme on peut multiplier deux matrices, on peut aussi calculer la puissance $n$-ième d'une matrice par $A^n = \underbrace{A \times A \times \ldots \times A \times A}_{n \text{ fois}}$.

1. Si $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ et $B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$, calculer $A \times B$ et $A^2$.
2. Montrer que pour toute matrice $A$, $A \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = A$.
3. Écrire un algorithme `produit_matrices_2_2` qui effectue le produit de deux matrices carrées d'ordre 2. Pour simplifier, la matrice $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ sera représentée par le tableau `[a, b, c, d]`. Ainsi, l'algorithme `produit_matrices_2_2` devra prendre en entrée deux tableaux `lst1` et `lst2` et renvoyer un tableau `lst`.
4. En s'inspirant de l'algorithme exponentiation rapide, écrire un algorithme `exp_bin_mat(a,n)` qui prend en entrée une matrice `a` et un entier naturel `n` et qui renvoie la matrice `a` à l'exposant `n`.
5. On rappelle que (ou on définit) la suite de Fibonacci $(F_n)_{n\in\mathbb{N}}$ est définie par :
$$\begin{cases} F_0 = 1 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n \ , \quad n \in \mathbb{N} \end{cases}$$
On montre que si $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, alors pour tout $n \geqslant 2$, $\quad A^n = \begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix}$.
En utilisant l'algorithme précédent, calculer $F_{48}$.

**Correction**

1. $A \times B = \begin{pmatrix} 1\times 5+2\times 7 & 1\times 6+2\times 8 \\ 3\times 5+4\times 7 & 3\times 6+4\times 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$

   $A^2 = \begin{pmatrix} 1\times 1+2\times 3 & 1\times 2+2\times 4 \\ 3\times 1+4\times 3 & 3\times 2+4\times 4 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$

2. Soit $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

   $A \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a\times 1+b\times 0 & a\times 0+b\times 1 \\ c\times 1+d\times 0 & c\times 0+d\times 1 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = A.$

3. 
```
def produit_matrices_2_2(lst1,lst2) :
    return [lst1[0]*lst2[0]+lst1[1]*lst2[2],lst1[0]*lst2[1]+lst1[1]*lst2[3],
            lst1[2]*lst2[0]+lst1[3]*lst2[2],lst1[2]*lst2[1]+lst1[3]*lst2[3]]
```

4. 
```
def exp_bin_mat(a,n) :
    res = [1,0,0,1]
    m = n
    puiss_a = a.copy()
    while (m != 0) :
        if (m%2 == 1) :
            res = produit_matrices_2_2(res,puiss_a)
        puiss_a = produit_matrices_2_2(puiss_a,puiss_a)
        m = m//2
    return res
```

5. On appelle `exp_bin_mat([1,1,1,0], 48)`.

   L'algorithme retourne `[7778742049, 4807526976, 4807526976, 2971215073]` donc $F_{48} = 7778742049$.

**Exercice 2.** ***Récursivité de de l'exponentiation « rapide »***

Codée en python

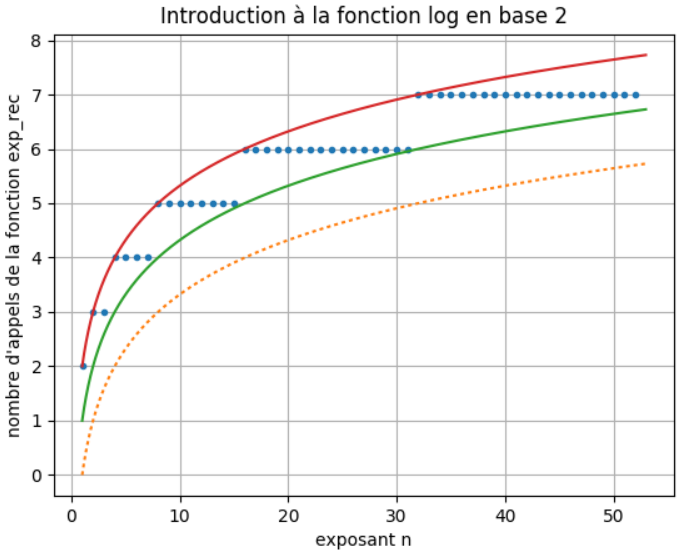
```
def exp_rec(a,n) :
    t_compteur[0] = t_compteur[0] + 1
    if n == 0 :
        return 1
    res = exp_rec(a,n//2)
    if n%2 == 0 :
        return res*res
    else :
        return res*res*a
```

On s'intéresse au nombre $C(n)$ de multiplications que la fonction récursive `exp_rec(a,n)` effectue lors de ses appels successifs.

⇒ Suivant la parité de $n$, le nombre de multiplications effectuées dans l'appel **d'une** fonction est 1 ou 2 (0 si $n = 0$).

⇒ On note $A(n)$ le nombre d'appels récursifs à l'appel de la fonction `exp_rec(a,n)`

| $n$ | 10 | 20 | 30 | 40 | 50 | 60 | 70 | ... |
|---|---|---|---|---|---|---|---|---|
| $A(n)$ | 5 | 6 | 6 | 7 | 7 | 7 | 8 | ... |




Approche de la fonction $\log_2$ par le graphique.

⇒ $\log_2(n) + 1 \leqslant A(n) \leqslant \log_2(n) + 2$

$\Rightarrow$ Évaluation de $C(n)$ à partir des exemples de nombre de multiplications :

| | | | | | |
|---|---|---|---|---|---|
| 31 | 15 | 7 | 3 | 1 | 0 |
| 2 | 2 | 2 | 2 | 2 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 64 | 32 | 16 | 8 | 4 | 2 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 43 | 21 | 10 | 5 | 2 | 1 | 0 |
| 2 | 2 | 1 | 2 | 1 | 2 | 0 |

On conjecture prouve que $C(n) = 2\,(A(n) - 1)$ et compte-tenu de l'encadrement de $A(n)$, on obtient

$$\log_2(n) \leqslant C(n) \leqslant 2\log_2(n) + 2$$

qui donne : $C(n) = \theta\,(\log_2(n))$.