

# TD - RECURSIVITE

## Exponentiation rapide

---

### Introduction et objectifs

Dans ce TD consacré à la récursivité, on étudie un algorithme permettant le calcul de  $a^n$  où  $a$  est un réel non nul et  $n$  un entier naturel. L'objectif principal du TD est une étude, à la fois expérimentale et théorique, de la complexité de l'algorithme proposée.

### Code Python

```
def fexpo(a,n):
    if n == 0:
        return 1
    else:
        q,r = divmod(n,2)
        prod = fexpo(a,q)
        prod = prod*prod
        if r == 1:
            prod = prod * a
        return prod
```

Remarque : la fonction « divmod » reçoit comme arguments deux entiers,  $\alpha$  et  $\beta$ , et renvoie deux autres entiers,  $q$  et  $r$ , correspondant respectivement au quotient et au reste de la division euclidienne de  $\alpha$  par  $\beta$ .

1. Expliquez le principe de l'algorithme ayant conduit à cette fonction Python.
2. Quelles sont les principales étapes du calcul de  $2^{53}$  ? Combien de multiplications sont requises pour ce calcul ? Combien de multiplications sont requises dans le calcul « naïf » ?

### Correction de l'algorithme

3. Une exécution de  $fexpo(a, n)$  se termine-t-elle toujours ?
4. Démontrer que l'exécution de  $fexpo(a, n)$  retourne bien la valeur de  $a^n$  (on démontrera le résultat en raisonnant par récurrence sur  $n$ ).

## Complexité de l'algorithme

Dans cette partie, on se donne un réel  $a$  non nul. On définit alors les suites  $M = (M_n)_{n \in \mathbb{N}}$  et  $A = (A_n)_{n \in \mathbb{N}}$  où, pour tout entier naturel  $n$ , les entiers  $M_n$  et  $A_n$  correspondent respectivement au nombre total de multiplications et au nombre total d'appels à la fonction `f expo` engendrés par un premier appel à cette fonction : `f expo ( a , n )` (cet appel initial ne sera donc pas comptabilisé au niveau de la suite  $A$ ). Ces quantités ne dépendent pas de la valeur du réel  $a$ .

Pour l'étude expérimentale des suites  $M$  et  $A$ , on écrira un programme mettant en œuvre la fonction `f expo` et incluant des variables globales `m` et `c` donnant, à la fin de l'exécution, le nombre total de multiplications effectuées et le nombre total d'appels à la fonction `f expo`.

### Etude de la suite $M$ .

5. Effectuer une étude expérimentale de la monotonie de la suite  $M$ .
6. Soit  $n$  un entier naturel non nul.  
On écrit  $n = 2q + r$  la division euclidienne de  $n$  par 2 et on définit la suite  $(u_n)_{n \in \mathbb{N}^*}$  par :  $\forall n \in \mathbb{N}^*, M_n = M_q + u_n$ .  
Préciser  $u_n$  en fonction de la parité de  $n$ .
7. Pour tout entier naturel  $p$ , calculer  $M_{2^p}$ ,  $M_{2^{p-1}}$  et  $M_{2^{p+1}}$ .  
Que peut-on en conclure sur la monotonie de  $M$  ?
8. Soit  $n$  un entier naturel non nul.  
Montrer qu'il existe un unique entier naturel  $p$  tel que :  $2^p \leq n < 2^{p+1}$ .  
Préciser  $p$ .
9. A l'aide d'une récurrence forte, montrer alors que l'on a :  $p + 2 \leq M_n \leq 2(p + 1)$ .
10. Dédurre de la question précédente que l'on a :  $M_n = O(\ln n)$ .

### Etude de la suite $A$ .

11. Effectuer une étude expérimentale de la monotonie de la suite  $A$ .
12. Pour  $n$  entier naturel non nul, on note  $n = 2q + r$  la division euclidienne de  $n$  par 2.  
Après avoir remarqué que l'on a  $A_n = A_q + 1$ , montrer, à l'aide d'une récurrence forte, que la suite  $A$  est croissante.
13. Pour tout entier naturel  $p$ , calculer  $A_{2^p}$ .
14. A l'aide de la question 8, déterminer un encadrement puis un équivalent de  $A_n$ .