DM 2 : Exponentiation rapide

À rendre pour le mardi 30 janvier, en complément de l'exercice 3 du TP 8.

Étant donné un flottant a et un entier $n \in \mathbb{N}^*$, on cherche à calculer numériquement a^n le plus efficacement possible. En effet, le calcul d'une puissance (notamment élevée) peut être beaucoup plus coûteuse en temps de calcul que les 4 opérations arithmétiques basiques $(+, -, \times, /)$. La commande « `a**n` » réalise ce calcul, mais il s'agit d'une « boite noire » : on ne dispose pas (aisément) de son code et donc on ne peut pas l'analyser. Le but de ce DM est de comparer deux algorithmes de calcul d'exponentiation.

On considère une première fonction :

```python
def naif(a,n):
    p = 1
    for k in range(n):
        p = p*a
    return p
```

1. Justifier que cet algorithme termine. Quel est sa complexité en fonction de la variable n ?

On s'intéresse maintenant à une autre fonction :

```python
def exporap(a,n):
    p = 1
    while n>0:
        if n%2==1:
            p = p*a
        a = a*a
        n = n//2
    return p
```

On note a_k, n_k, p_k les valeurs des variables a, n, p de l'algorithme ci-dessus à la fin de la k-ième itération de la boucle `while`. Par convention, a_0, n_0, p_0 sont les valeurs des variables avant l'entrée dans la boucle.

2. On suppose qu'on exécute `exporap(5,84)`. Faites un tableau qui exprime les valeurs de p_k, a_k, n_k à chaque itération (laisser de la place pour une cinquième colonne). On ne calculera pas explicitement les puissances de 5, on pourra se contenter de les noter 5^q avec $q \in \mathbb{N}$. Vérifier que la valeur retournée est bien 5^{84}.
3. Sur la colonne restante, écrire le nombre d'opérations élémentaires effectuées par l'itération numéro k de la boucle `while`. En déduire le nombre total d'opérations élémentaires effectuées par l'instruction `exporap(5,84)`. Comparer avec `naif(5,84)`.
4. Démontrer la terminaison de l'algorithme `exporap`.

On s'intéresse maintenant à la complexité de `exporap`, qu'on exprimera par rapport à l'argument n. On fixe donc $n \in \mathbb{N}^*$ et on cherche à déterminer le nombre d'opérations de `exporap(a,n)` dans le pire cas.

5. Est-ce qu'il y a un argument a qui conduit à un pire cas ? Si oui, lequel ?
6. On note c_n le nombre d'opérations élémentaires effectuées par `exporap` (dans l'éventuel pire cas) en fonction de n. Exprimer c_{2n} en fonction de c_n. De même, exprimer c_{2n+1} en fonction de c_n.
7. On suppose dans cette question que $n = 2^q$ avec $q \in \mathbb{N}$. Déterminer c_n et l'exprimer en fonction de $\log_2(n)$, i.e. le logarithme en base 2 de n.

Dans le cas général, en écrivant n en base 2, on peut montrer que la complexité recherchée est en $\log_2(n)$, ce qui est bien plus avantageux que celle de l'approche naïve. Cette technique est utilisée par la commande « `a**n` », entre autres.