UNIVERSITÉ DE RENNES I
U.F.R. DE MATHÉMATIQUES

UNIVERSITÉ DE RENNES 1

Licence 2 Mathématiques et Applications **2018-2019**
U.E. Outils Informatiques B

TD 1 - Calculs de puissances

Exercice 1 - Codage de Fibonacci

On rappelle que la suite de Fibonacci est définie par les conditions initiales $f_0 = 0$, $f_1 = 1$ et la relation de récurrence $f_{n+2} = f_{n+1} + f_n$.

1. Rédiger en Python une fonction nommée `pgf` prenant en argument un entier $n \geq 1$ et renvoyant le plus grand terme f_k de la suite de Fibonacci vérifiant $f_k \leq n$.
   Faire tourner le code "à la main" dans le cas où n=50.

   Tout entier $n \geq 1$ peut être décomposé en somme de termes distincts de la suite de Fibonacci. Par exemple, $50 = 34 + 8 + 5 + 3 = f_9 + f_6 + f_5 + f_4$. Cependant, pour que cette décomposition soit unique on doit ajouter la contrainte suivante : on n'utilise pas f_0 et f_1 et on s'interdit d'avoir dans la décomposition de n deux termes consécutifs de la suite de Fibonacci. Par exemple, la décomposition précédente de 50 ne convient pas, mais $50 = 34 + 13 + 3 = f_9 + f_7 + f_4$ convient. Ce résultat constitue le théorème de Zeckendorf.

2. Montrer (par récurrence) l'existence d'une telle décomposition pour tout entier $n \geq 1$. Nous admettrons son unicité.

   Le codage de Fibonacci est une représentation des entiers naturels non nuls fondée sur cette décomposition. Si
   $$n = \sum_{i=0}^{k-1} d_i f_{i+2}$$
   vérifie les conditions :
   $$\forall i \in \{0, \cdots, k-1\}, d_i \in \{0,1\}, d_i d_{i+1} = 0 \text{ et } d_{k-1} = 1$$
   alors l'entier n sera représenté par la chaîne de caractères $d_0 d_1 d_2 \cdots d_{k-1}$.
   Par exemple, l'entier 50 sera représenté par la chaîne de caractères `"00100101"` puisque
   $$50 = 0f_2 + 0f_3 + 1f_4 + 0f_5 + 0f_5 + 1f_7 + 0f_8 + 1f_9$$

3. Rédiger en PYTHON une fonction nommée `decode` qui prend en paramètre un code de Fibonacci et qui renvoie l'entier n représenté par ce code. Par exemple, `decode('00100101')` devra retourner 50.
4. Rédiger en PYTHON la fonction inverse nommée `code` : celle-ci prend en paramètre un entier strictement positif et retourne le code de Fibonacci de ce nombre. Par exemple, `code(50)` retournera la chaîne de caractères `'00100101'`.
5. Dans cette question on s'intéresse au calcul de x^n lorsque x et n sont des entiers non nuls en cherchant à minimiser le nombre de multiplications utilisées (et en s'interdisant bien entendu l'usage de l'opérateur $**$).
   (a) Rédiger en PYTHON une première solution utilisant $n-1$ multiplications (on définira une fonction prenant deux arguments `x` et `n` et retournant la valeur de x^n).
   (b) Si $k \geq 2$, montrer par recurrence que le calcul de x^{f_k} peut être réalisé à l'aide de $k-2$ multiplications.
   (c) En déduire une fonction `puissance` qui calcule x^n lorsque n est représenté par son code de Fibonacci. Par exemple, `puissance(2, '00100101')` devra retourner 1125899906842624 (c'est la valeur de 250).