Devoir maison n°8, à rendre le vendredi 14 avril 2023

Problème : Décomposition de Zeckendorf

Partie I : Suite et nombres de Fibonacci

On considère la suite de Fibonacci $(F_n)_{n\in\mathbb{N}}$, définie par $F_0 = 0$, $F_1 = 1$ et, pour tout entier $n \in \mathbb{N}$,

$$F_{n+2} = F_{n+1} + F_n. \quad (\star)$$

Les nombres F_n ($n \in \mathbb{N}$) de cette suite sont appelés les nombres de Fibonacci.

1. **Quelques propriétés de la suite de Fibonacci :**
   (a) Montrer que, pour tout $n \in \mathbb{N}^{\geqslant 1}$, $F_n > 0$ et $F_{n+1} > 0$.
   (b) Montrer que la suite $(F_n)_{n\in\mathbb{N}}$ est strictement croissante à partir du rang 2.
   (c) Déterminer, pour tout $n \in \mathbb{N}$, une expression explicite de F_n.
   (d) En déduire que $(F_n)_{n\in\mathbb{N}}$ diverge et que $\lim\limits_{n\to+\infty} F_n = +\infty$.
2. **Calcul des premiers nombres de Fibonacci avec Python :**
   (a) Ecrire une fonction Python `fibo(n)` prenant comme argument $n \in \mathbb{N}$ et renvoyant la valeur de F_n. Pour construire cette fonction, utiliser $(\star)$.
   (b) Que réalise le script suivant ?

```
L=[]
for k in range(20):
    L.append(fibo(k))
print(L)
```

   Exécuter ce script et vérifier qu'il renvoie :

`[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181].`

3. Ecrire une fonction `recherche(x,L)` prenant comme arguments d'entrées :
   - un entier naturel `x`
   - une liste `L` déjà triée dans l'ordre croissant, dont le premier élément est inférieur ou égal à `x` et le dernier est strictement supérieur à `x`

   et qui renvoie le plus grand élément de la liste `L` qui soit inférieur ou égal à `x`.

Partie II : Décomposition de Zeckendorf et algorithme glouton

On s'intéresse dans cette partie au théorème suivant qui énonce que tout nombre $n \in \mathbb{N}^{\geqslant 1}$ s'écrit, de façon unique, comme une somme de nombre de Fibonacci F_c ($c \in \mathbb{N}^{\geqslant 2}$) distincts et non consécutifs.
Autrement dit, de façon formelle :

Théorème de Zeckendorf :

Pour tout $n \in \mathbb{N}^{\geqslant 1}$, il existe un unique entier $k \in \mathbb{N}^{\geqslant 1}$ et un unique k-uplets d'entiers $(c_1, \ldots, c_k)$, vérifiant :

- $c_1 \geqslant 2$
- pour tout $i \in [\![1; k-1]\!]$, $c_i + 1 < c_{i+1}$

tels que :

$$n = \sum_{i=1}^{k} F_{c_i}.$$

Cette décomposition s'appelle la **décomposition de Zeckendorf** du nombre n.

Exemples :

- $n = 4$ se décompose en : $4 = 1 + 3 = F_2 + F_4$. Dans ce cas : $k = 2$ et $(c_1, c_2) = (2, 4)$;
- $n = 17$ se décompose en : $17 = 1 + 3 + 13 = F_2 + F_4$. Dans ce cas : $k = 3$ et $(c_1, c_2, c_3) = (2, 4, 7)$;

4. On rappelle que la liste des premiers termes de la suite $(F_n)_{n\in\mathbb{N}}$ a été donnée dans la question 1.
   (a) En remarquant que $6 = 1 + 2 + 3 = F_2 + F_3 + F_4$ et aussi $6 = 1 + 5 = F_2 + F_5$, donner la décomposition de Zeckendorf de 6 et justifier votre choix.
   (b) Donner la décomposition de Zeckendorf de 35.
   (c) Donner la décomposition de Zeckendorf de 130.
5. **Algorithme Python renvoyant la décomposition de Zeckendorf d'un entier** $n \in \mathbb{N}^{\geqslant 1}$

   La fonction Python ci-dessous prenant pour argument d'entrée un entier $n \in \mathbb{N}^{\geqslant 1}$ et renvoie la décomposition de Zeckendorf de cet entier.

```
def Zeckendorf(n):
     i=0
     L=[fibo(i)]
     while L[-1]<=n:
          i=i+1
          L.append(fibo(i))
     k=n
     T=[]
     while k>0:
          f=recherche(k,L)
          T.append(f)
          k=k-f
     return T
```

   (a) Expliquer la méthode proposée par cet algorithme pour déterminer la décomposition de Zeckendorf d'un entier $n \in \mathbb{N}^{\geqslant 1}$. Cet algorithme étant composée de deux boucles conditionnelles, vous expliquerez l'intérêt de chacune d'elles ainsi que les critères d'arrêts associés.
   (b) En quoi l'algorithme précédent est-il un algorithme glouton ?

⋆ ⋆ ⋆ **Fin du sujet** ⋆ ⋆ ⋆