Corrigé de l'interrogation d'Informatique numéro 1

Sujet inspiré de l'épreuve 2015 de mathématiques 1 de E3A PC

On s'intéresse aux nombres de Fibonacci définis par récurrence par $F_0 = 0$, $F_1 = 1$ et

$$\forall n \in \mathbb{N}, \quad F_{n+2} = F_{n+1} + F_n$$

1. Écrire en langage Python une fonction `fib(n)` permettant le calcul de F_n .
2. On admettra que ces nombres vérifient les relations suivantes pour tout $n \in \mathbb{N}^*$:

$$F_n = 2F_{\frac{n}{2}-1}F_{\frac{n}{2}} + \left(F_{\frac{n}{2}}\right)^2 \text{ si } n \text{ est pair et } , F_n = \left(F_{\frac{n-1}{2}+1}\right)^2 + \left(F_{\frac{n-1}{2}}\right)^2 \text{ sinon}$$

Vérifier cette propriété pour $n = 6$.

Compléter l'écriture de la fonction `fibb(n)` ci-dessous permettant de calculer F_n , on commencera au dernier `else` sur la copie en rajoutant les instructions manquantes.

```
def fibb(n):
    if n<=1 :
        return n
    else :
        if n%2 == 0:
            a=fibb(n//2)
            b=fibb(n//2-1)
            return a*(2*b+a)
        else : ...
```

3. (a) Soit $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Pour tout $n \in \mathbb{N}^*$, exprimer les coefficients de A^n au moyen des nombres de Fibonacci F_{n+1} , F_n et F_{n-1} .
- (b) Ecrire en langage Python une fonction `fibbb(n)` permettant le calcul de F_n en utilisant le calcul de A^n (dont le calcul doit intervenir dans le programme).
- (c) Montrer que pour tout $n \in \mathbb{N}^*$, on a $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.
4. (a) Montrer que la série de terme général $\frac{(-1)^n}{F_{n+1}F_n}$ ($n \in \mathbb{N}^*$) est absolument convergente et

exprimer $S = \sum_{n=1}^{+\infty} \frac{(-1)^n}{F_{n+1}F_n}$ au moyen de $L = \lim_{p \rightarrow +\infty} \frac{F_p}{F_{p+1}}$.

Utiliser des sommes télescopiques

- (b) On admettra que pour tout

$$(\star) \quad n \leq n_0 \in \mathbb{N}^*, \left| S - \sum_{n=1}^{n_0} \frac{(-1)^n}{F_{n+1}F_n} \right| \leq \frac{1}{F_{n_0+1}F_{n_0+2}}$$

Soit $\epsilon > 0$, écrire un programme Python permettant de donner une valeur approchée de L avec une précision inférieure à ϵ à l'aide de la propriété $(\star)$

Corrigé :

1. (a) On a `fib(n)` définie par :

```
def fib(n):
    if n<2:
        return n
    else:
        return fib(n-1)+fib(n-2)
```

- (b) On a $F_6 = 2F_2F_3 + F_3^2$. Or, $F_2 = F_1 + F_0 = 1$ et $F_3 = F_2 + F_1 = 1 + 1 = 2$. On en déduit que $F_6 = 2 \times 2 \times 1 + 2^2 = 4 + 4 = 8$.

```
def fibb(n):
    if n<=1 :
        return n
    else :
        if n%2 == 0:
            a=fibb(n//2)
            b=fibb(n//2-1)
            return a*(2*b+a)
        else :
            a=fibb((n-1)//2+1)
            b=fibb((n-1)//2)
            return a**2+b**2
```

2. (a) On a pour tout $n \in \mathbb{N}$, $\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = A \times \begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix}$ donc $\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = A^n \times \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$.

On a :

```
def fibbb(n):
    A=np.array([[1,1],[1,0]])
    Acc=np.array([[1,0],[0,1]])
    for k in range(n):
        Acc=np.dot(A,Acc)
    return Acc[0,1]
```

- (b) On raisonne par récurrence. Pour $n = 1$, on a bien $F_2F_0 - F_1^2 = 1 \times 0 - 1^2 = -1 = (-1)^1$. Supposons maintenant la relation vraie au rang $n \geq 1$, alors on a :

$$F_{n+1}F_{n-1} - F_n^2 = (F_n + F_{n-1})F_{n-1} - F_n^2 = F_nF_{n-1} + F_{n-1}^2 - F_n^2 (*)$$

Or par hypothèse de récurrence, on a $F_n^2 = F_nF_{n-2} - (-1)^n = F_nF_{n-2} + (-1)^n$. d'où

$$(*) = F_nF_{n-1} + F_nF_{n-2} - F_n^2 + (-1)^n = F_n(F_{n-1} + F_{n-2}) - F_n^2 + (-1)^n = (-1)^n$$

On en déduit que l'hérédité est vérifiée.

- (c) Pour tout $n \geq 3$, $F_n \geq n$ par récurrence. On a ainsi pour tout $n \geq 3$, $|\frac{(-1)^n}{F_{n+1}F_n}| \leq \frac{1}{n^2}$. On en déduit que la série associée est absolument convergente.

De plus, pour tout $n \in \mathbb{N}^*$, $\frac{(-1)^n}{F_{n+1}F_n} = \frac{F_{n+1}F_{n-1} - F_n^2}{F_{n+1}F_n} = \frac{F_{n-1}}{F_n} - \frac{F_n}{F_{n+1}}$ donc la série est télescopique et la somme de la série vaut $S = \frac{F_0}{F_1} - L = -L$.

3. Le programme suivant donne une approximation de L à ϵ près d'après (*).

```
def approche(eps):
    puis = -1
    S = 0
    Fn, Fn1 = 1,1
    while Fn*Fn1 < 1/eps :
```

```
S += puis/(Fn*Fn1)
puis = -puis
Fn, Fn1 = Fn1 , Fn+Fn1
return -S
```