ISM MPSI/PCSI, Informatique
Année 2024/2025

TP Informatique 17

Exercice 1

On considère *la suite de Fibonacci* $(u_n)_n$ définie par $u_0 = 0$, $u_1 = 1$ et

$$\forall n \in \mathbb{N} \qquad u_{n+2} = u_{n+1} + u_n$$

Il s'agit d'une suite récurrente linéaire d'ordre deux à coefficients constants. Par une récurrence immédiate, la suite est à valeurs dans $\mathbb{N}$. Par ailleurs, on peut démontrer l'égalité

$$\forall n \in \mathbb{N} \qquad u_n = \frac{1}{\sqrt{5}}\left(\varphi^n - \frac{(-1)^n}{\varphi^n}\right) \quad \text{avec} \quad \varphi = \frac{1+\sqrt{5}}{2}$$

1. Écrire, en utilisant l'expression du terme général de la suite $(u_n)_n$, une fonction `fibo1(n)` d'argument n entier qui renvoie la valeur de u_n. On effectuera un unique calcul de puissance. En particulier, le terme $(-1)^n$ ne sera pas calculé par exponentiation.
2. Afficher les dix premières valeurs de la suite $(u_n)_n$. On pourra utiliser la liste par compréhension `[fibo1(n) for n in range(10)]`. Que constate-t-on sur la nature du résultat ? Que renvoie `fibo1(2000)` ?
3. Écrire une fonction `expo(x,n)` d'arguments `x` un flottant, `n` un entier et qui renvoie la valeur de x^n calculé par exponentiation rapide. Tester la fonction pour différentes valeurs de x et n.
4. Écrire une fonction `fibo2(n)` d'argument n entier qui renvoie la valeur de u_n. On effectuera un unique calcul de puissance par exponentiation rapide. Vérifier que les dix premières valeurs de la suite $(u_n)_n$ calculée à l'aide de `fibo2` sont conformes à celles obtenues avec `fibo1`.
5. Écrire une fonction `fibo3(n)` d'argument n entier qui renvoie la valeur de u_n en la calculant à l'aide d'une boucle en ne manipulant que des objets de type `int`. Vérifier que les dix premières valeurs de la suite $(u_n)_n$ calculée à l'aide `fibo3` sont conformes à celles obtenues avec `fibo1` et `fibo2`.
6. Déterminer le seuil à partir duquel les réponses entre `fibo1` et `fibo3` ont un écart supérieur ou égal à un.
7. Récupérer le fichier `TP17_EX01.py` sur le site de la classe, y copier-coller les fonctions `fibo1`, `fibo2` et `fibo3` et lancer le script (qui peut durer quelques instants). Celui-ci affiche un graphique des temps d'exécution des trois fonctions précédentes en fonction de l'argument n. Commenter les résultats obtenus.

Exercice 2

Dans l'exercice précédent, on a utilisé l'exponentiation rapide mais sur du calcul flottant donc inexact. Dans cet exercice, on se propose de réaliser un calcul exact en ne manipulant que des objets de type `int` et en utilisant à nouveau l'exponentiation rapide. La suite $(u_n)_n$ désigne la suite de Fibonacci définie précédemment. On pose

$$\forall n \in \mathbb{N} \qquad \mathrm{X}_n = \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix}$$

On rappelle la définition de la multiplication matricielle avec

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} = \begin{pmatrix} aa'+bc' & ab'+bd' \\ ca'+dc' & cb'+dd' \end{pmatrix}$$

On observe $$\forall n \in \mathbb{N} \qquad \mathrm{X}_{n+1} = \begin{pmatrix} u_{n+1} \\ u_{n+2} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{=\mathrm{A}} \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix} = \mathrm{A}\mathrm{X}_n$$

La suite $(\mathrm{X}_n)_n$ vérifie une relation géométrique et comme pour une suite réelle ou complexe, on a

$$\forall n \in \mathbb{N} \qquad \mathrm{X}_n = \mathrm{A}^n \mathrm{X}_0$$

Ainsi, le calcul du u_n pour n entier peut être obtenu à partir de l'exponentation matricielle A^n. La matrice A sera saisie par `np.array([[0,1],[1,1]],dtype=object)`. La matrice identité, élément neutre pour le produit matriciel, sera saisie par `np.array([[1,0],[0,1]],dtype=object)`. Le produit de deux matrices `M` et `N` sera réalisé par l'instruction `np.dot(M,N)`.

1. Écrire une fonction `expo(X,n)` d'arguments `X` une matrice (format `ndarray`), `n` un entier et qui renvoie la matrice X^n calculé par exponentiation rapide. Tester la fonction pour la matrice A et différentes valeurs de n.
2. Écrire une fonction `fibo4(n)` d'argument n entier qui renvoie la valeur de u_n à l'aide d'une exponentiation rapide matricielle. Vérifier que les dix premières valeurs de la suite $(u_n)_n$ calculée à l'aide `fibo4` sont conformes à celles obtenues avec `fibo3`.
3. Récupérer le fichier `TP17_EX02.py` sur le site de la classe, y copier-coller les fonctions `fibo3`, `fibo4` et lancer le script (qui peut durer quelques instants). Celui-ci affiche un graphique des temps d'exécution des deux fonctions précédentes en fonction de l'argument n. Commenter les résultats obtenus.