

TP - complexité

K. Bertet – J.L. Guillaume – A. Huchet

L'objectif de cette séance est d'effectuer une comparaison des trois algorithmes utilisés pour calculer la suite de Fibonacci afin de voir si la complexité théorique des algorithmes garde son sens en pratique.

1. Programmation des trois algorithmes

exercice 1. Le premier objectif est de programmer les trois algorithmes suivants qui calculent tous le n-ème nombre de Fibonacci (https://fr.wikipedia.org/wiki/Suite_de_Fibonacci). On néglige ici le fait que $Fibo(0)=0$.

Version 1 :

```
Fibo1 (n) -- n entier naturel non nul
  si n ≤ 2 retourner 1
  retourner Fibo1(n-1)+Fibo1(n-2)
```

Version 2 :

```
Fibo2 (n) -- n entier naturel non nul
  a=1
  b=1
  pour i allant de 3 à n
    b = a+b
    a = b-a
  retourner b
```

Version 3 (nécessite d'écrire une fonction pour faire le produit de deux matrices 2x2)

```
PuissanceMatrice(A,n) -- calcule la n-ème puissance de la matrice A
  si n = 1
    retourner A
  si n est pair
    retourner PuissanceMatrice(A, n/2) * PuissanceMatrice(A, n/2)
  retourner A * PuissanceMatrice(A, (n-1)/2) * PuissanceMatrice(A, (n-1)/2)

Fibo3 (n) -- n entier naturel non nul
  si n ≤ 2 retourner 1
  A = PuissanceMatrice( $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ , n-1)
  retourner A(1,1)
```

exercice 2. Vérifier que vos trois implémentations donnent le même résultat pour plusieurs valeurs de n.

exercice 3. Montrer que Fibo3 calcule effectivement les nombres de Fibonacci.

2. Comparaison des temps de calcul

exercice 4. Calculer la complexité et le temps d'exécution réel des trois algorithmes (à vous de trouver une solution en python, utilisant par exemple `time.clock`) pour faire une courbe affichant les temps de calcul pour n allant de 1 à x (x à choisir de manière raisonnée pour les 3 algorithmes). Vérifiez que la complexité théorique et le temps d'exécution en pratique ont bien la même allure.

3. Calcul des tous les nombres de Fibonacci

Si le temps le permet en séance, sinon après la séance.

exercice 5. On souhaite maintenant calculer tous les nombres de Fibonacci pour i allant de 1 à n sans modifier les algorithmes précédents (juste avec une boucle for). Refaites les courbes et indiquez les complexités correspondantes.

exercice 6. Peut-on faire mieux si on peut modifier les trois algorithmes ? Expliquez-comment, modifiez les algorithmes et indiquez les complexités.