

TD 2

Exercices d'application

Savoirs et compétences :

- Alg – C15 : Récursivité : avantages et inconvénients.

Exercice 1 – Fonction mystère

D'après ressources de C. Lambert. On donne la fonction suivante :

```

■ Python
def mystere(L):
    """
    Ceci est la fonction mystère, saurez-vous trouver
    son but ?
    Entrée :
        * L(list) : liste de nombres entiers ou réels
    Sortie :
        * ???
    """
    n = len(L)
    if n==0 :
        return (None)
    elif n==1 :
        return (L[0])
    else :
        x = mystere(L[0:n-1])
        if x<L[-1] :
            return (x)
        else :
            return (L[-1])

```

Question 1 Sans coder la fonction, déterminer le résultat de l'instruction `print(mystere([14, 20, 3, 16]))` ? Vous pourrez représenter de façon graphique l'empilement et le dépilement de la pile d'exécution.

Question 2 D'après vous quel est le but de cette fonction ?

Question 3 Programmer la fonction et tester l'instruction précédente. Sur plusieurs exemples, vérifiez la conjecture faite à la question précédente.

Question 4 Question subsidiaire. – Montrer que la propriété suivante est une propriété d'invariance :

\mathcal{P} : l'algorithme retourne le plus petit élément de la liste de taille k , s'il existe.



- Il faudra montrer que l'algorithme se termine au moyen d'un variant de boucle.
- Il faudra montrer que \mathcal{P} est une propriété d'invariance.

Exercice 2 – Palindrome...

D'après ressources de C. Lambert. On souhaite réaliser une fonction *miroir* dont le but est de retourner le «miroir» d'une chaîne de caractères. Par exemple le résultat de *miroir*("miroir") serait "riorim".

Question 1 Programmer la fonction *miroir_it* permettant de répondre au problème de manière itérative.

Question 2 Programmer la fonction *miroir_rec* permettant de répondre au problème de manière récursive.

Question 3 Que renvoie la fonction si la chaîne de caractère est "Eh ! ça va la vache" ?

Question 4 Évaluer la complexité algorithmique de chacune des deux fonctions.

Exercice 3 – Suite de Fibonacci

D'après ressources de C. Lambert.

On définit la suite de Fibonacci de la façon suivante :

$$\forall n \in \mathbb{N}, \begin{cases} u_0 = 0, u_1 = 1 \\ u_{n+2} = u_n + u_{n+1} \end{cases}$$

Question 1 Définir la fonction *fibonacci_it* permettant de calculer u_n par une méthode itérative. Évaluer la complexité algorithmique de l'algorithme.

Question 2 Définir la fonction `fibonacci_rec` permettant de calculer u_n par une méthode récursive « intuitive ». Évaluer la complexité algorithmique de l'algorithme.

Question 3 Observer comment passer du couple (u_n, u_{n+1}) au couple (u_{n+1}, u_{n+2}) . En déduire une autre méthode récursive pour calculer le n^{e} terme de la suite de Fibonacci. Évaluer la complexité algorithmique de l'algorithme.

Exercice 4 – Faisons des Bulles

D'après les ressources de Mmes SEMBELY et VERDIER

Les fractales sont des objets mathématiques fondés sur des figures géométriques se répétant à l'infini via un processus itératif.

"Une fractale est un objet irrégulier, dont l'irrégularité est la même à toutes les échelles et en tous les points" Adrien DOUADY - mathématicien français

Ce type de structure se retrouve également dans la nature : architecture des côtes maritimes, ramifications nerveuses, nuages, galaxies ...

Etant donné leur structure, il est très intéressant d'utiliser la récursivité pour visualiser des ensembles fractals.

On utilisera des tableaux **numpy** pour représenter les

points. C'est plus pratique que les listes python pour faire les calculs vectoriels.

- Si a et b représentent respectivement les points (x, y) et (x', y') alors $a + b$ représente le point $(x + x', y + y')$.

- Si r est un réel et a représente le point (x, y) alors $r * a$ représente le point $(r x, r y)$.

- Si a et b sont des tableaux **numpy** alors `dot(a, b)` représente le produit matriciel $a \times b$ (si ce produit est possible). La fonction **dot** est une fonction numpy.

Question 1 Ecrire une fonction `cercle` d'arguments a et r qui trace le cercle de centre le point a et de rayon r (supposé strictement positif).

Question 2 Ecrire une fonction récursive `bubble` d'arguments a, r, n . Elle effectuera la construction pour un nombre n d'étapes en ayant pour figure de base le cercle de centre a et de rayon r (supposé strictement positif).

Question 3

