Série N° 2 (suite)

Exercice 1

Ecrire une fonction **maximum(t)** qui retourne la plus grande valeur contenue dans un tableau t de n nombres entiers.

1) Calculer sa complexité dans le pire des cas.
2) Calculer sa complexité dans le meilleure des cas.
3) Calculer sa complexité en moyenne des cas.

Exercice 2

On se propose d'écrire une fonction renverse(t) pour renverser le contenu d'un tableau. Par exemple si on passe en paramètre à la fonction renverse le tableau t=[1, 2, 3, 4], après l'appel t contiendra [4, 3, 2, 1].
La fonction suivante n'atteint pas l'objectif fixé :

1) Le vérifier en observant le résultat de l'appel sur le tableau **t = [1, 2, 3, 4]**.
Quel est le contenu de t après chaque tour de boucle ?
2) Proposer une modification de la fonction renverse pour renverser effectivement le tableau.
3) Calculer la complexité de la fonction reverse dans le pire des cas
4) Un tableau est dit "palindrome" si on lit la même suite de nombres en le parcourant de gauche à droite ou de droite a gauche. Par exemple, [6, 2, 7, 4, 7, 2, 6] est un tableau "palindrome", [2, 0, 0, 2] également.
Ecrire une fonction **estPalindrome1(t)** qui retourne True ou False suivant que t est un tableau "palindrome" ou pas (on peut utiliser la fonction reverse).
5) Donner la complexité de la fonction estPalindrome1(t) dans le pire des cas.
6) Ecrire une autre version **estPalindrome2(t)** sans utiliser la fonction reverse. Analyser sa complexité

Exercice 3

On se donne une matrice m contenant p lignes et q colonnes dont les éléments sont uniquement 0 et 1. On veut trouver des entiers $i_0 \in [0, p]$ et $j_0 \in [0, q]$ tels que $i_0 + j_0$ soit maximum et $\forall i \in [0, i_0[$, $\forall j \in [0, j_0[$, $m_{i,j} = 1$.

Pour simplifier on suppose la matrice non vide : p<= 1 et q<= 1 (il est inutile de les vérifier).

1) Quel résultat doit–on obtenir si $m_{0,0} = 0$?

Les étudiantes Donadora, Euphrasie et Frédégonde proposent chacune le code suivant :

Code de Donadora :

```
def donadora (m) :
    (p,q) = (len(m) , len (m[0]))
    (i0, j0) = (0, 0)
    for i in range (p+1) :
        for j in range (q+1) :
            res = True
            for u in xrange (i) :
                for v in xrange (j) :
                    if m[u,v] == 0 :
                        res = False
            if res and i+j > i0+j0 :
                (i0,j0) = (i,j)
    return (i0,j0)
```

Code d'Euphrasie :

```
def euphrasie (m) :
    (p,q) = (len(m),len(m[0]))
    (i0, j0) = (0, 0)
    while m[i0, j0] and i0 < p :
        i0 += 1
    while m[i0-1, j0] and j0 < q :
        j0 += 1
    return (i0, j0)
```

Code frédégonde

```
def fredegonde (m) :
    (p,q) = (len(m),len(m[0]))
    (i0, imax, j0) = (p, p, 0)
    j = 0
    while j < q :
        i = 0
        while i < imax and m[i,j] :
            i += 1
        imax = i
        j += 1
        if i+j > i0+j0 :
            (i0,j0)=(i,j)
    return (i0,j0)
```

2) Parmi les codes de Donadora, d'Euphrasie et de Frédégonde, l'un est faux. Lequel ? Donner un exemple de matrice pour laquelle on obtiendrait un résultat erroné.
3) Donner les complexités des deux codes restants à chaque fois sous la forme O(f(p, q)).