E3A PSI 2018 mathématiques 1

—

Corrigé

Exercice 2 (original 4)

1. La fonction suivante retourne le résultat demandé pour toute liste d'entiers naturels non vide :

```
def maxi(L):
    res = L[0]
    for k in range(1,len(L)):
        if L[k] > res:
            res = L[k]
    return res
```

2. La fonction suivante retourne le résultat demandé pour toute liste d'entiers naturels :

```
def ind(L):
    res = []
    for k in range(len(L)):
        if L[k] != 0:
            res.append(k)
    return res
```

3. Par exemple (avec la même restriction qu'en **1**) :

```
def nb_oc(L):
    M = maxi(L)+1
    res = [0]*M
    for k in range(len(L)):
        res[L[k]] += 1
    return res
```

4.1. Lors de l'appel `nb_oc(L)` avec la fonction `nb_oc` telle que définie en **3**, la liste `L` est parcourue une première fois pour déterminer `M`, puis une seconde fois pour le remplissage de la liste `res`. Donc la liste `L` est parcourue deux fois en tout.

4.2. Le nombre `n = 2` trouvé en **4.1** est indépendant de `M`.

5.1. On trouve $L_1 = [1,4,3,3,2,2,3,1,1,0]$, $L_2 = [1,4,3,3,2,2,3,1,1,0] = L_1$, et donc par récurrence immédiate, $L_k = L_1$ pour tout $k \geqslant 1$.

5.2. La liste B proposée ne peut pas être la liste L_1 d'une suite de Robinson puisque par construction dans une telle liste, les valeurs $i_r, \ldots, i_2, i_1$ présentes dans les cases d'indice impair vont en décroissant.

5.3. Les listes L_0 correspondant à une telle liste L_1 sont toutes les listes constituées de deux « 4 » et d'un « 0 » (et de rien d'autre). Il y a donc trois possibilités : $[0,4,4]$, $[4,0,4]$ ou $[4,4,0]$.

5.4. Par exemple (avec la même restriction qu'en **1**) :

```
def rob(A,n):
    L = A
    for _ in range(n):
        T = nb_oc(L)
        I = ind(T)
        L = []
        for i in I[::-1]:  # I[::-1] est la liste I renvers\'ee, i.e. [i_r,...,i_1].
            L.append(T[i])
            L.append(i)
    return L
```