

## Chap. 3 Tableaux

## Thème 2

# Occurrences d'une valeur dans un tableau

## Sommaire du chapitre 3

- ▷ Retour sur les tableaux
- ▷ Recherche dans un tableau
- ▷ Matrices : tableaux de tableaux
- ▷ **Occurrences d'une valeur dans un tableau**
- ▷ Organisation des tableaux en mémoire
- ▷ Tableaux et références

## I Recherche de toutes les occurrences d'une valeur dans un tableau

Lorsque l'on cherche toutes les occurrences d'une valeur (ou du maximum) dans un tableau, on renvoie en général une `list` Python contenant l'indice de chaque occurrence. Par exemple, sur le tableau `[1, 7, 4, 15, 7, 18, 7]`, une recherche de la valeur 7 renvoie `[1, 4, 6]`, alors qu'une recherche de la valeur 2 renvoie `[]`. On pourrait adapter le code de `recherche_tableau` en créant un tableau `r` avant la boucle, en remplaçant `return i` par `r.append(i)` et en remplaçant `return -1` par `return []`. Mais il est beaucoup plus élégant d'utiliser une compréhension de liste `[ i for i, v in enumerate(t) if v ==x ]`.

## II Recherche de toutes les occurrences du maximum dans un tableau

Pour calculer toutes les occurrences du maximum, on peut procéder en deux étapes : calculer la valeur maximale, puis calculer les occurrences de cette valeur par compréhension. Ou on peut directement calculer toutes les occurrences en adaptant le code de `recherche_tableau` :

```
1 def occurrences_max_tableau(t) :  
2     """ Entrée : tableau t non vide  
3     Renvoie : la liste des indices des valeurs maximales dans t """  
4     r = []  
5     val_max = t[0] # on suppose t != []  
6     for i in range(len(t)) :  
7         if t[i] == val_max :  
8             r.append(i)  
9         elif t[i] > val_max :  
10            val_max = t[i]
```

```
11         r = [i] # réinitialise r
12     return r
13
14 occurrences_max_tableau(['b', 'c', 'a', 'c']) # [1, 3]
```

### III Recherche de toutes les occurrences d'une valeur (ou du maximum) dans une matrice

Pour chercher de même toutes les occurrences d'une valeur (ou du maximum) dans une matrice, on peut adapter le code de `recherche_matrice` :

```
1 def occurrences_matrice(x, m) :
2     """ Entrée : valeur x, matrice m bien formée, non vide
3     Renvoie : la liste des coordonnées des occurrences de x dans m """
4     r = [ ]
5     for i in range(len(m)) :
6         for j in range(len(m[i])) :
7             if m[i][j] == x :
8                 r.append((i, j))
9     return r
```

Ainsi `occurrences_matrice(5, [[2, 5], [1, 4], [5, 5], [6, 0]])` renvoie `[[0, 1], [2, 0], [2, 1]]`. On peut aussi obtenir ce résultat avec une compréhension imbriquée :  
`[(i, j) for i, ligne in enumerate(m) for j, v in enumerate(ligne) if v == x]`.