

## Chap. 3 Tableaux

## Thème 2

# Recherche 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

Lorsqu'un tableau `t` n'est pas trié, le seul moyen de trouver si un élément est présent dans le tableau `t` est de parcourir `t` :

```
1 def recherche_tableau(x, t) :  
2     """ Renvoie le même résultat que l'expression Python : x in t """  
3     for y in t :  
4         if x == y :  
5             return True  
6     return False # si on atteint ce point, x n'est pas dans t  
7 rechercher_tableau('a', ['c', 'b', 'a']) # True
```

En général, on ne s'intéresse pas seulement à savoir si `x` est présent dans le tableau, mais aussi à trouver les indices auxquels `x` est présent.

Dans l'exemple ci-dessus, on interrompt la boucle dès que l'élément `x` est trouvé. Selon la position de la valeur cherchée, la boucle s'arrête dès le premier élément, ou parcourt tout le tableau. Par contre, si on cherche à renvoyer toutes les occurrences de `x`, on doit toujours parcourir le tableau jusqu'au bout avant de renvoyer le résultat puisqu'une interruption prématurée ne peut pas garantir qu'aucun élément non parcouru ne fasse partir du résultat. Il en va de même lorsque l'on veut calculer l'élément maximum du tableau.

```
1 def max_tableau(x, t) :  
2     """ Renvoie le même résultat que la fonction Python : max(t) """  
3     max_temporaire = t[0] # on suppose len(t)>0  
4     for y in t :  
5         if y >= max_temporaire :  
6             max_temporaire = y  
7     return max_temporaire
```

Lorsque le tableau est trié, on peut chercher les éléments plus efficacement, comme nous le verrons au chapitre suivant. Par exemple, on peut trouver l'indice de n'importe quel entier `x` sur le tableau `[1, 3, 4, 5, 6, 8, 12]` en trois comparaisons maximum.