

## COMPÉTENCE 5 : Comment rechercher une occurrence dans une liste, triée ou non ?

### Exercice 5.1 : Générer une liste aléatoire (★)

---

Écrire une fonction `genereListeAlea(n,max)` qui renvoie une liste de  $n$  entiers appartenant à  $[[0, max]]$ .

Écrire une fonction `genereListeTrie(n)` qui renvoie une liste de  $n$  entiers triés de 0 à  $(n - 1)$ .

### Exercice 5.2 : Combien de fois ? (★)

---

1. Écrire une fonction `compte(elm,liste)` qui renvoie le nombre d'occurrences de `elm` dans `liste`, à l'aide de l'iterable `range()` et d'une structure inconditionnelle.
2. Le parcours de liste est-il total ou partiel ?

### Exercice 5.3 : Combien de foisBis ? (★)

---

Même question mais sans utiliser l'iterable `range()`.

### Exercice 5.4 : Combien de foisTer ? (★★)

---

Écrire une fonction `compte3(elm,liste)` qui renvoie le nombre d'occurrences de `elm` dans `liste`, à l'aide d'une structure conditionnelle.

### Exercice 5.5 : Présent, m'sieur ! Mais c'est abuser ! (★)

---

Écrire une fonction booléenne `present(elm,liste)` qui renvoie `True` si `elm` est dans `liste`, `False` sinon, à l'aide d'une structure conditionnelle. L'interruption de boucle est ici autorisée quoique fortement déconseillée...

### Exercice 5.6 : Présent, m'sieur 2 ! (★★)

---

Écrire une fonction booléenne `present(elm,liste)` qui renvoie `True` si `elm` est dans `liste`, `False` sinon, à l'aide d'un booléen et d'une structure conditionnelle.

### Exercice 5.7 : Et par dichotomie ? (★★)

---

Soit  $L$  une liste déjà triée. Écrire une fonction `chercheDicho(elm,liste)` qui renvoie l'indice de position de `elm` dans `liste`, et  $(-1)$  sinon, à l'aide d'une recherche par dichotomie.

Question subsidiaire : quelle est la complexité des algorithmes des questions 3.1 à 3.6 ? 3.7 doit-il être classé à part ?

*Partie 1 – Questions de cours*

- 1 Rappeler le principe de la recherche dichotomique dans une liste d'entiers. Quel intérêt présente cette méthode ?
- 2 La mise en œuvre d'une recherche dichotomique est-elle possible sur une liste de couples d'entiers ? de chaînes de caractères ?

*Partie 2 – Étude d'une fonction dichotomie*

Voici le code d'une fonction Python élaboré pour tester par dichotomie si un entier  $x$  se trouve dans une liste d'entiers `liste` :

```
(1) def dichotomie(liste, x):
(2)     # Pré-conditions: x est un entier, liste est une
(3)     # liste d'entiers triée dans l'ordre croissant
(4)     n = len(liste)
(5)     if n == 0:
(6)         return False
(7)     g, d = 0, n-1
(8)     while d-g > 0:
(9)         m = (g+d)//2
(10)        if liste[m] >= x:
(11)            d = m
(12)        else:
(13)            g = m
(14)    return liste[g] == x
```

- 3 Pour quelles raisons ne remplace-t-on pas la précondition de la ligne (3) par un appel à une fonction qui trierait la liste `liste` dans l'ordre croissant ?
- 4 Justifier que le prédicat  $\mathcal{P}$  : « l'entier  $x$  apparaît dans la sous-liste `liste[g:d+1]` des éléments de `liste` d'indices  $g$  à  $d$  » est préservé à chaque tour de la boucle `while`.
- 5 Il s'avère que la fonction `dichotomie` ne termine pas. Donner un exemple où la fonction boucle.
- 6 Indiquer sans justification la ou les corrections à apporter pour que la fonction `dichotomie` termine, tout en restant correcte.
- 7 Justifier que la fonction corrigée termine et est correcte.