

Exercices : Algorithmes de parcours sur les tableaux

Exercice 1

1. Écrire ci-contre la fonction recherche du cours implémentant l'algorithme de recherche d'une occurrence d'une valeur `val` parmi les éléments d'un tableau `tab`. Cette fonction doit renvoyer :
 - le plus petit indice `index` tel que `tab[index] = val` si `val` est présent dans `tab`
 - -1 sinon

```
def recherche(val, tab) :  
    ...  
    ...
```

2. S'agit-il d'un parcours par indice ou par élément ?
3. Si la valeur recherchée est présente aux indices 1234, 784 989 et 3 785 654, quel sera l'indice retourné par la fonction recherche ?
4. Dans un tableau de taille 100 000 000, où doit être positionnée la valeur cherchée pour que la recherche soit rapide ?
5. Dans un tableau de taille 100 000 000, où doit être positionnée la valeur cherchée pour que la recherche soit lente ?
6. Si la valeur cherchée n'est pas présente parmi les éléments du tableau, la recherche est-elle rapide ou lente ? Justifier en quelques mots.
7. Sur un tableau `T` de taille 100 000 000 et sur une machine donnée, `recherche(7, T)` a pris 31 ms. Combien de temps prendra `recherche(8, T)` sur la même machine ?

Exercice 2

1. En vous inspirant de l'algorithme de calcul d'une moyenne des éléments d'un tableau, écrire ci-contre une fonction `compter_occurences` permettant de compter le nombre d'occurrences de `val` dans le tableau `tab` (autrement dit, compter combien de fois `elt` est présent dans `tab`).

En particulier se poser la question : " Vais-je faire un parcours par élément ou par indice ?"

```
def compter_occurences(val, tab):  
    compteur = 0  
    ...
```

Exemples d'assertions vérifiées par la fonction `compter_occurences`

```
T = [1, 7, 4, 7, 5, 3, 8, 9]  
assert(compter_occurences(7, T)==2)  
assert(compter_occurences(4, T)==1)  
assert(compter_occurences(77, T)==0)
```

2. Est-ce que `compter_occurences` fonctionne avec un tableau de chaînes de caractères ?
3. Est-ce qu'on parcourt systématiquement tout le tableau ?

Le coût est-il par conséquent tout le temps linéaire ou seulement dans le pire des cas ?

4. Pour cette question on admet que le temps d'exécution est linéaire comme le coût (*).
 Sur un tableau d'entiers T1 de taille 1 000 000 et sur une machine donnée, compter_occurrences(10, T1) a pris 150 ms.
 Sur un tableau d'entiers T2 de taille 15 000 000 et sur la même machine, combien de temps prendra approximativement compter_occurrences(1000, T2)?

(*) : en réalité on ne passe pas aussi simplement de la complexité algorithmique au temps d'exécution sur machine.

Exercice 3

En reprenant les deux exercices ci-dessus, comparer brièvement les deux graphiques ci-dessous :

- expliquer le point commun
- expliquer la différence

