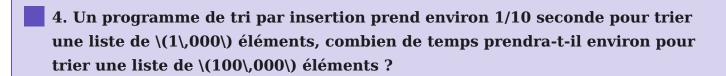
Devoir 4 : Les tris - Diviser pour régner



1. Q.C.M.



- 1. On applique l'algorithme du tri par sélection à la liste [8,12,6,19], après la première étape, le contenu de la liste sera :
 - a) [12,8,6,19]
 - b) [6,12,8,19]
 - c) [19,12,6,9]
 - d) Aucune des propositions ci-dessus
- 2. On applique l'algorithme du tri par insertion à la liste [9,11,7,16], quel sera le contenu de la liste après le premier échange ?
 - a) [11,9,7,16]
 - b) [9,11,16,7]
 - c) [9,7,11,16]
 - d) Aucune des propositions ci-dessus
- 3. L'algorithme du tri par sélection a une complexité :
 - a) logarithmique
 - b) linéaire
 - c) quadratique
 - d) exponentielle



- a) 1 seconde
- b) 10 secondes
- c) 100 secondes
- d) 1000 secondes

2. D'aprés exercice BAC



Exo

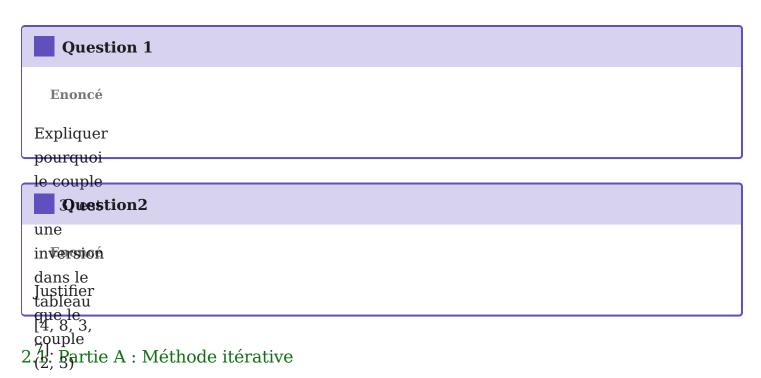
Cet exercice traite de manipulation de tableaux, de récursivité et du paradigme « diviser pour régner ».

Dans un tableau Python d'entiers tab, on dit que le couple d'indices ($\dot{\cdot}$) forme une inversion lorsque $\dot{\cdot}$ et tab[i] > tab[j]. On donne ci-dessous quelques exemples.

- Dans le tableau [1, 5, 3, 7], le couple d'indices (1,2) forme une inversion car 5> 3. Par contre, le couple (1,3) ne forme pas d'inversion car 5<7. Il n'y a qu'une inversion dans ce tableau.
- Il y a trois inversions dans le tableau [1, 6, 2, 7, 3], à savoir les couples d'indices (1, 2), (1, 4) et (3, 4).
- On peut compter six inversions dans le tableau [7, 6, 5, 3]: les couples d'indices (0, 1), (0, 2), (0, 3), (1, 2), (1, 3) et (2, 3).

On se propose dans cet exercice de déterminer le nombre d'inversions dans un tableau quelconque.

Questions préliminaires



Le but de cette partie est d'écrire une fonction itérative nombre_inversion qui renvoie le nombre d'fiversions dans un tableau. Pour cela, on commence par écrire une fonction fonction1 qui sera ensure utilisée pour écrire la fonction nombre_inversion.

Question A.1

Enoncé

On donne la fonction suivante.

Script Python def fonction1(tab, i): nb_elem = len(tab) cpt = 0 for j in range(i+1, nb_elem): if tab[j] < tab[i]: cpt</pre>

a. Indiquer ceque renvoie lafonction1(tab,i) dans les cassuivants.

return cpt

+= 1

```
• Cas n°1:
tab = [1, 5,
3, 7] et i =
0.
```

```
• Cas n°2:
tab = [1, 5,
3, 7] et i =
1.
```

```
• Cas n°3:
tab = [1, 5,
2, 6, 4] et i
= 1.
```

b. Expliquer ce

que permet de

déterminer

tQuestion.A.2

Enoncé

En utilisant la fonction précédente, écrire une fonction nombre_inversion(tab) qui prend en argument un tableau et renvoie le nombre d'inversions dans ce tableau. On donne ci-dessous les résultats attendus pour certains appels.

Texte Question A.3

nombre inversions([1, 5, 7])

Quelle est

l'ordrerdenversions([1,

grandeur])

de³la

2 complexitéversions([7,

Leen⁶temps

l'algorithme

obtenu?

Aucune

justification

n'est

attendue.

e récursive

econcevoir une version récursive de la fonction nombre_inversion. ctions auxiliaires.

Question B.1

Enoncé

Donner le nom d'un algorithme de tri ayant

une

complexité meilleure

que

quadratique.

Dans la

suite de cet

exercice, on

suppose

qu'on

dispose

d'une

fonction

tri(tab) qui

prend en

argument

un tableau

et renvoie

un tableau

contenant

les mêmes

éléments

rangés dans

l'ordre

croissant.

Question B.2

Enoncé

Écrire une
fonction
moitie_gauche(tab)
qui prend en
argument un
tableau tab et
renvoie un
nouveau tableau
contenant la
moitié gauche de
tab. Si le nombre
d'éléments de tab

est impair,

Danélémente dun suppose qu'on dispose de la fonction moitie_droite (tab) qui renvoie la moitié droite sans léfé pent du veille u.

dans cette partie gauche. On donne cidessous les résultats attendus pour certains appels.

Script Python >>> moitie_gauche([]) [] >>> moitie_gauche([4, 8, 3]) [4, 8] >>> moitie_gauche ([4, 8, 3, 7]) [4, 8]

Question B.3

Enoncé

On suppose qu'une fonction nb_inv_tab(tab1, tab2)a été écrite. Cette fonction renvoie le nombre d'inversions du tableau obtenu en mettant bout à bout les tableaux tab1 et tab2, à condition que tab1 et tab2 soient triés dans l'ordre croissant.
On donne ci-dessous deux exemples d'appel de cette fonction :

% Script Python

```
>>> nb_inv_tab([3, 7, 9], [2, 10])
3
>>> nb_inv_tab([7, 9, 13], [7, 10, 14])
3
```

En utilisant la fonction nb_inv_tab et les questions précédentes, écrire une fonction récursive nb_inversions_rec(tab) qui permet de calculer le nombre d'inversions dans un tableau. *
Cette fonction renverra le même nombre que nombre_inversions(tab) de la partie A. On

procédera de la façon suivante :

- Séparer le tableau en deux tableaux de tailles égales (à une unité près).
- Appeler récursivement la

fonction
nb_inversions_rec
pour compter le
nombre
d'inversions dans
chacun des deux
tableaux.

- Trier les deux tableaux (on rappelle qu'une fonction de tri est déjà définie).
- Ajouter au nombre d'inversions précédemment comptées le nombre renvoyé par la fonction nb_inv_tab avec pour arguments les deux tableaux triés.