

TP N° 3 : DIVISER POUR RÉGNER

Syntaxe Python :

- Suppression/remplacement d'éléments d'une liste :
 - suppression de l'élément d'indice 2 de la liste $L=[1, 2, 3, 4, 5, 6]$:
 $L[2:3]=[]$
 $\# L=[1, 2, 3, 4, 5, 6]$ devient $[1, 2, 4, 5, 6]$
 - suppression des éléments d'indices 1,2 et 3 de la liste $L=[1,2,3,4,5,6]$:
 $L[1:4]=[]$
 $\# L=[1, 2, 3, 4, 5, 6]$ devient $[1, 5, 6]$
 - remplacement des éléments d'indices 2 et 3 de la liste $L=[1,2,3,4,5,6]$ par la valeur 10
 $L[2:4]=[10]$
 $\# L=[1, 2, 3, 4, 5, 6]$ devient $[1, 10, 6]$
 - remplacement du slice $L[2:5]$ de la liste $L=[1,2,3,4,5,6]$ par $t=[20,30,40]$
 $L[2:5]=[20, 30, 40]$
 $\#$ ou bien $L[2:5]=t$
 $\# L=[1, 2, 3, 4, 5, 6]$ devient $[1, 2, 20, 30, 40, 6]$
 - remplacement du slice $L[2:3]$ de la liste $L=[1,2,3,4,5,6]$ par $t=[20,30,40]$
 $L[2:3]=[20, 30, 40]$
 $\#$ ou bien $L[2:3]=t$
 $\# L=[1, 2, 3, 4, 5, 6]$ devient $[1, 2, 20, 30, 40, 4, 5, 6]$
- Mesure du temps d'exécution :


```
from time import*
t1=clock()
#actions
...
t2=clock()-t1
```

Exercice 1 : Tri rapide

- Version non « en place »¹ du tri rapide :
 Écrire une fonction `partition(deb, fin, L)` qui modifie la liste L de telle sorte que :
 - tous les éléments à gauche du pivot soient inférieurs au pivot.
 - le pivot prenne sa place définitive ; on prendra pour pivot $L[deb]$.
 Cette fonction renverra l'indice du pivot dans la liste L modifiée.

deb	deb+1	fin-1	fin
4 =pivot	3	5	5	2	1	2

deb...	indice pivot	...fin
<pivot	4	>=pivot

¹Un algorithme est dit « en place » s'il n'utilise qu'un nombre très limité de variables et s'il modifie directement la structure qu'il est en train de trier. Le plus souvent un tel algorithme est plus difficile à écrire.

En déduire la fonction `TriRapide` associée et la tester avec la fonction `test_tri` (du fichier `sort.py`)

2. En utilisant l'algorithme vu en cours, écrire une fonction `TriRapide1(deb, fin, L)` qui trie sur place un tableau entre les indices `deb` et `fin` passés en paramètres. Tester ensuite avec la fonction `test_tri`.
3. Mesurer les temps d'exécution des divers tris pour une même liste : tri par insertion, tris rapides, tris fusions.

Exercice 2 : Couples croissants

Étant donnés 2 entiers a et b , on s'intéresse aux couples croissants de l'intervalle $[a, b]$. Par exemple, les couples croissants de $[2, 5]$ sont $(2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)$.

Écrire les fonctions suivantes :

1. `f(a, b)` qui renvoie le nombre de solutions
2. `f1` qui affiche les couples solutions
3. `f2` qui place dans une liste les couples solutions

Évidemment on mettra en pratique le schéma « Diviser pour régner ».

Donner ensuite une version purement itérative de ces mêmes fonctions.