

## TP 7 | Algorithmes de tri

### Exercice 1 : Tri par insertion (avec liste auxiliaire)

Remarque préliminaire : interdiction d'utiliser `pop`, `insert`, `remove`... `append` est évidemment autorisé

1) Créer une fonction `f_deplace_terme(L, Ind)` qui, en supposant que la liste `L` est triée pour les indices inférieurs à `Ind`, déplace le terme d'indice `Ind` à la bonne place dans la partie déjà triée.

Vérifiez bien les trois exemples :

```
>>> L=[0,2,4,5,6,2,7]   >>> L = [4,5,7,1]   >>> L = [4,5,1,7]
>>> f_deplace_terme(L,5) >>> f_deplace_terme(L,3) >>> f_deplace_terme(L,2)
>>> L
[0, 2, 2, 4, 5, 6, 7]   >>> L
[1, 4, 5, 7]           >>> L
[1, 4, 5, 7]
```

2) Créer une fonction `f_insertion(L, Val)` qui insert une valeur `val` dans la liste `L` déjà triée et qui la déplace afin que `L` soit triée en ordre croissant.

3) Mettre en place une fonction `f_tri_insertion_aux(L)` qui renvoie une liste triée de `L` par la méthode du cours sur le tri par insertion avec liste auxiliaire.

4) Vérifier que le tri est stable avec l'exemple `[3, 2, 1, 1, 0]` et améliorer la fonction de déplacement dans le cas où il ne le serait pas.

### Exercice 2 : Tri sélection

Le tri sélection d'une liste `L` consiste à réaliser l'algorithme suivant :

- Etape 0 :
  - o Détermination de l'indice  $i_0$  du minimum de `L`
  - o Echange de `L[0]` et `L[i0]`
- Etape 1 :
  - o Détermination de l'indice  $i_1$  du minimum de `L[1:]`
  - o Echange de `L[1]` et `L[i1]`
- Etc...
- Etape n-2 :
  - o Détermination de l'indice  $i_{n-1}$  du minimum de `L[n - 2:]`
  - o Echange de `L[n - 2]` et `L[in-2]`

1) Proposer une fonction  $Ind\_min(L, i)$  renvoyant l'indice python du premier minimum de  $L$  à partir de l'indice  $i$  inclus.

Remarque : L'utilisation de la fonction `min` de python est interdite.

2) Proposer une fonction  $f\_tri\_selection(L)$  qui réalise ce tri en place (sans création de nouvelle liste).

3) Proposer une fonction  $f\_tri\_selection\_rec(L)$  qui réalise ce tri en place de manière récursive.

Remarques :

- On veillera à appeler  $L$  entièrement, et non une sous liste de  $L$  du genre  $L[...]$  qui serait traitée comme une nouvelle liste et qui ne respecte donc pas le principe « en place » ;
- On pourra ajouter un argument optionnel  $i=0$ .

4) En étudiant le tri de la liste  $[3, 2, 1, 1, 0]$ , et en modifiant  $Ind\_min$  pour qu'elle renvoie l'indice du dernier minimum, discuter de la condition nécessaire à la stabilité de ces tris.

## Exercice 3 : Tri fusion

1) Mettre en place une fonction  $f\_diviser\_1(L)$  qui renvoie les deux listes ( $L_1$  et  $L_2$ ) issues de la scission (séparation en deux) en son milieu de  $L$ .

2) Mettre en place une fonction  $f\_fusion\_ordonnee(L_1, L_2)$  qui renvoie une liste ordonnée issue des différents termes de  $L_1$  et  $L_2$  supposée déjà triées.

Vérifiez :

```
>>> f_fusion_ordonnee_1([5],[3,8])  
[3, 5, 8]
```

```
>>> f_fusion_ordonnee_1([3,4],[1,2])  
[1, 2, 3, 4]
```

```
>>> f_fusion_ordonnee_1([1,2],[3,4])  
[1, 2, 3, 4]
```

```
>>> f_fusion_ordonnee_1([1,3,5],[2,4,6])  
[1, 2, 3, 4, 5, 6]
```

3) Mettre en place une fonction  $f\_tri\_fusion(L)$  qui renvoie une liste triée de  $L$  par la méthode du tri fusion avec listes auxiliaires par récursivité.

4) Vérifier que le tri est stable avec l'exemple  $[3, 2, 1, 1, 0]$  et améliorer la fonction  $f\_fusion\_ordonnee(L_1, L_2)$  dans le cas où il ne le serait pas.

## Exercice 4 : Tri rapide

Nous supposons que le pivot est choisi naïvement comme la première valeur de la liste.

1) Mettre en place une fonction  $f\_combine(Listes)$  où  $Listes$  est une liste de listes de taille quelconque, qui renvoie la liste combinée de gauche à droite des listes de  $Listes$ .

Exemple :

```
>>> f_combine([[1,2,3],[4],[5,6],[5,4]])
[1, 2, 3, 4, 5, 6, 5, 4]
```

2) Mettre en place une fonction  $f\_diviser\_2(L)$  qui renvoie une liste de 3 listes de la forme  $[L_1, [Pivot], L_2]$  où  $L_1$  est la liste des éléments inférieurs ou égaux au pivot et  $L_2$  la liste des éléments strictement supérieurs à celui-ci.

Remarque : ne pas inclure le premier terme (pivot) dans  $L_1$ ...

Exemple :

```
>>> f_divise([5,2,9,4,7,3,6])
[[2, 4, 3], [5], [9, 7, 6]]
```

3) Mettre en place une fonction  $f\_tri\_rapide\_aux(L)$  qui renvoie une liste triée de  $L$  avec listes auxiliaires par récursivité en utilisant les deux fonctions précédentes.

Essayez :

```
>>> L = [1,2,3,1.0]
>>> f_tri_rapide_aux(L)
[1, 1.0, 2, 3]
```

4) Si votre tris n'est pas stable, améliorer la fonction  $Divise$  afin d'obtenir la stabilité.