

Dernière mise à jour	Informatique	Denis DEFAUCHY – Site web
27/01/2022	8 - Tris	TD 8-1 – Tri par insertion

Informatique

8 Tris

TD 8-1 *Tri par insertion*

Dernière mise à jour	Informatique	Denis DEFAUCHY – Site web
27/01/2022	8 - Tris	TD 8-1 – Tri par insertion

Exercice 1: Tri par insertion

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

Question 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 – On veillera à ce que sa complexité soit en $O(1)$ dans le meilleur des cas et que l'algorithme s'exécute « en place » (pas de création de liste, pas de return)

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                    >>> L                    >>> L
[0, 2, 2, 4, 5, 6, 7]    [1, 4, 5, 7]    [1, 4, 5, 7]
```

Cette fonction pourra être utilisée pour les deux méthodes de tri par insertion suivantes.

Tri avec liste auxiliaire

Question 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 (pas de return)

Question 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

Tri en place

Question 4: Mettre en place une fonction $f_tri_insertion_en_place(L)$ qui trie la liste L par la méthode du cours sur le tri par insertion en place

Stabilité

Vérifiez :

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

Question 5: Si votre tri n'est pas stable, améliorez la fonction de déplacement afin de rendre les tris stables

Dernière mise à jour	Informatique	Denis DEFAUCHY – Site web
27/01/2022	8 - Tris	TD 8-1 – Tri par insertion

BONUS - Version récursive

A ne réaliser que si vous êtes en avance !

On envisage une version récursive de ce tri qui fonctionne de la sorte :

- Cas de base au rang 0 : la liste ne contient aucun terme ($n=0$), elle est renvoyée
- Cas général au rang n :
 - Création de la liste L_g des $n-1$ termes de gauche de L et de son dernier terme t
 - Tri récursif de L_g , créant la liste triée L_{gt}
 - Insertion de t à la bonne place dans L_{gt}
 - Renvoi de la liste L_{gt} triée

Question 6: Proposer une fonction `f_tri_insertion_aux_rec(L)` qui trie L de manière récursive avec listes auxiliaires