

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
07/04/2022	8 - Tris	TD 8-3 – Tri fusion

# Informatique

## 8 Tris

***TD 8-3***  
***Tri fusion***

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
07/04/2022	8 - Tris	TD 8-3 – Tri fusion

## Exercice 1: Tri fusion

### *Tri avec liste auxiliaire*

**Question 1:** Mettre en place une fonction  $f\_fusion\_ordonnee\_1(L_1, L_2)$  non récursive 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])      >>> f_fusion_ordonnee_1([3,4],[1,2])
[3, 5, 8]                                [1, 2, 3, 4]

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

**Question 2:** Mettre en place une fonction  $f\_tri\_fusion(L)$  qui renvoie une liste triée de  $L$  par la méthode du cours sur le tri fusion avec listes auxiliaires par récursivité

Il est possible de créer une fusion ordonnée récursive.

**Question 3:** Mettre en place une fonction  $f\_fusion\_ordonnee\_2(L_1, L_2)$  récursive réalisant le même travail que la fusion ordonnée non récursive ( $L_1$  et  $L_2$  sont triées)

Attention : dès que les listes sont trop grandes, on dépasse la capacité de la pile d'exécution et cette fonction récursive fait boguer python : **maximum recursion depth exceeded in comparison**. On gardera donc la fusion 1 dans ce tri.

Vérifiez :

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

**Question 4:** Si vos tris ne sont pas stables, avec les fusions ordonnées 1 ou 2, améliorez celles-ci afin d'obtenir la stabilité

**La suite de ce TP ne sera réalisée que si vous avez de l'avance**

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
07/04/2022	8 - Tris	TD 8-3 – Tri fusion

## *Tri en place*

Nous allons maintenant mettre en place un algorithme de tri fusion en place.

Pour cela, il faut créer une fonction de fusion ordonnée en place sur  $L$ , voici une manière de procéder (je ne dis pas tout quand même) :

- Définir deux indices  $i_1 = i$  et  $i_2 = j$
- Tant qu' $i_1 < j$  et  $i_2 < k$  :
  - o Echanger si besoin  $L[i_1]$  et  $L[i_2]$  en enlevant un terme que l'on insérera au bon endroit dans  $L$  (cf remarques ci-dessous)
  - o Mettre à jour le/les indice(s) qui le nécessitent
- Réfléchir s'il faut faire quelque chose quand  $i_1 = j$  ou  $i_2 = k$

Remarques :

- $L.pop(i)$  pour enlever le terme d'indice  $i$  de  $L$
- $L.insert(i,t)$  pour insérer le terme  $t$  à l'indice  $i$  de  $L$ , les autres étant décalés
- Ces fonctions étant optimisées sous Python, on supposera qu'elles sont de complexité en temps  $O(1)$ .
- On pourrait réaliser l'algorithme avec la seule condition  $i_1 < i_2$  mais cela conduirait à traiter des termes déjà triés dans certains cas où  $i_2 = k$  et  $i_1 < i_2$

**Question 5: Mettre en place une fonction  $f\_fusion\_ordonnee\_ep(L, i, j, k)$  non récursive qui trie en place dans  $L$  de manière ordonnée ses éléments dans l'intervalle d'indices Python  $[i, k[$  sachant que les deux sous listes dans  $[i, j[$  et  $[j, k[$  sont triées**

Essayez :

```
>>> L = [1,4,8,9,10,2,4,5,9,11]
>>> fusion_ordonnee_ep(L,0,5,10)
>>> L
[1, 2, 4, 4, 5, 8, 9, 9, 10, 11]
```

**Question 6: Mettre en place une fonction  $tri\_fusion\_ep(L)$  récursive qui réalise le tri fusion de  $L$  – On définira une sous fonction récursive  $rec(L, i, j)$**

Vérifiez :

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

**Question 7: Si votre tri n'est pas stable, améliorez la fonction de fusion ordonnée afin d'obtenir la stabilité**

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
07/04/2022	8 - Tris	TD 8-3 – Tri fusion

Encore du temps ? On propose une nouvelle version de la fusion ordonnée en place qui procède de la manière suivante : Pour chaque terme dans la partie L2 (de  $j$  à  $k$ ), on le descend par échanges successifs jusqu'à la bonne position dans la partie de gauche L1

**Question 8: Mettre en place une fonction  $f\_fusion\_ordonnee\_ep\_bis(L, i, j, k)$  non récursive qui trie en place dans  $L$  de manière ordonnée ses éléments dans l'intervalle d'indices Python  $[i, k[$  sachant que les deux sous listes dans  $[i, j[$  et  $[j, k[$  sont triées**

Essayez :

```
>>> L = [1,4,8,9,10,2,4,5,9,11]
>>> fusion_ordonnee_ep_bis(L,0,5,10)
>>> L
[1, 2, 4, 4, 5, 8, 9, 9, 10, 11]
```

ATTENTION : il y a bien deux conditions à mettre dans le while. Si vous n'y avez pas porté attention, vous aurez un problème dans cet exemple :

```
>>> L = [0,2,5,6,1,2]
>>> fusion_ordonnee_ep_bis(L,2,4,6)
>>> L
[0, 2, 1, 2, 5, 6]

>>> L = [1,2,0]
>>> fusion_ordonnee_ep_bis(L,0,1,3)
>>> L
[0, 1, 2]
```