



Préparation à l'épreuve pratique NSI

12

Fusion de deux listes triées

Nom :

Note :

/ 20

Classe :

1 Spécification du programme à réaliser

1.1 Description générale

À partir de deux listes triées, on peut facilement construire une liste triée comportant les éléments issus de ces deux listes (leur *fusion*). Le plus petit élément de la liste à construire est soit le plus petit élément de la première liste, soit le plus petit élément de la deuxième liste. Ainsi, on peut construire la liste élément par élément en retirant tantôt le premier élément de la première liste, tantôt le premier élément de la deuxième liste.

On considère deux tableaux triés A [1,...,a] et B [1,...,b]. L'objectif consiste à construire un tableau trié qui contient exactement les éléments des tableaux A et B

Entrées	Sorties
Les tableaux A [] et B []	Un tableau trié contenant tous les éléments de A et de B.

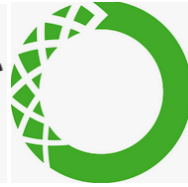
1.2 Pseudo code de l'algorithme¹

```

fonction fusion(A[1, ..., a], B[1, ..., b])
    si A est le tableau vide
        renvoyer B
    si B est le tableau vide
        renvoyer A
    si A[1] ≤ B[1]
        renvoyer A[1] ⊕ fusion(A[2, ..., a], B)
    sinon
        renvoyer B[1] ⊕ fusion(A, B[2, ..., b])
    
```

Le symbole \oplus désigne ici la concaténation de tableaux. (L'opération + avec des listes en python)

¹ Voir https://fr.wikipedia.org/wiki/Tri_fusion



1.3 Amélioration

- On traitera le cas où les deux listes données en paramètre ne contiennent aucun élément. La fonction retournera la valeur **None** dans ce cas.

2 Codage et mise en œuvre

2.1 Script de l'exercice



 NSI-PROG-012-Fusion-Liste-Trie.py

2.2 Fonction à compléter

```
## Votre fonction à réaliser
def fusion_de_deux_listes(liste_
```

```
    )
    return ....
```

Conseil de codage : pour sélectionner tous les éléments d'une liste python sauf le premier faire le slicing de la manière suivante :

```
>>> liste_A = [-13, -8, -2, 10, 41, 52, 56]
```

```
>>> liste_C = liste_A[1:]
```

```
>>> liste_C
[-8, -2, 10, 41, 52, 56]
```

Point particulier : vous noterez pour la première fois l'utilisation d'une fonction qui s'appelle elle-même. C'est la programmation récursive.

2.3 Résultats attendus

```
>>> (executing lines 1 to 99 of "NSI-PROG-006-Fusion-Liste-Trie-corrige.py")
```

```
Liste_A : [-13, -8, -2, 10, 41, 52, 56]
```

```
Liste_B : [-10, 8, 52, 84, 250]
```

```
Listes fusionnées : [-13, -10, -8, -2, 8, 10, 41, 52, 52, 56, 84, 250]
```

```
Liste_A : []
```

```
Liste_B : [-10, 8, 52, 84, 250]
```

```
Listes fusionnées : [-10, 8, 52, 84, 250]
```

```
Liste_A : [-13, -8, -2, 10, 41, 52, 56]
```

```
Liste_B : []
```

```
Listes fusionnées : [-13, -8, -2, 10, 41, 52, 56]
```

```
Liste_A : []
```

```
Liste_B : []
```

```
Listes fusionnées : None
```

