

# Informatique 2ème année, CPBX, TD2

Paul Dorbec

## 1 Tri Fusion

Dans ce TD, on va mettre en place le tri fusion. Je vous rappelle que vous pouvez utiliser `genere_liste(n)` disponible dans le fichier `test_dichot.py` pour générer une liste aléatoire. À l'url <https://docs.python.org/3/tutorial/datastructures.html>, vous trouverez une description complète des opérations sur les listes. En particulier, vous aurez besoin de

- l'instruction `x=L.pop(0)` qui permet de retirer un élément en tête d'une liste `L` et de l'enregistrer dans la variable `x`.
- l'instruction `L.append(x)` qui permet d'ajouter l'élément `x` à la fin de la liste `L` (en position `len(L)`).

**Question 1.1** *Écrivez une fonction `fusion(L1,L2)` qui prend en entrée deux listes triées et produit une nouvelle liste triée regroupant l'ensemble des éléments de `L1` et `L2`. On ne supposera pas que les deux listes sont de même taille.*

Il existe une solution relativement simple en python, qui permet en particulier de ne pas avoir à gérer des arrondis. Tout d'abord, on transforme la liste à trier en une liste de listes de un seul élément. `[3,1,2]` est réécrite en `[[3],[1],[2]]`. La liste  $\mathcal{L}$  sur laquelle on travaille est désormais une liste de listes, toutes triées. La méthode consiste maintenant à prendre les deux premières listes de  $\mathcal{L}$ , à les fusionner en une nouvelle liste triée, et à ajouter cette nouvelle liste triée à la fin de  $\mathcal{L}$ . Dès lors que  $\mathcal{L}$  ne contient plus qu'un élément, cet élément est la liste de départ triée.

**Question 1.2** *Implémentez le tri fusion en suivant ces indications.*

**Question 1.3** *Que se passe-t-il si dans l'algorithme précédent, au lieu d'ajouter la nouvelle liste en fin de liste  $\mathcal{L}$  (avec `L.append(x)`), on l'ajoute en début de liste (avec `L.insert(0,x)`) ? Quelle est la complexité du tri ainsi formé ?*

**Question 1.4** *Si vous avez fini le tri fusion, vous pouvez implémenter le tri à bulle, assez facile à réaliser.*