ITC – TD

Algorithmes de tris – 1

On se propose au cours de quelques séances de TD de tester et approfondir notre étude des tris classiques vus en cours. Comme dans le cours, on utilisera des numpy array afin d'avoir un comportement naturel pour les tris en place récursifs.

> Un fichier est fourni avec le TD à préparer : il contient des en-têtes de fonctions et procédures, à modifier. Ainsi vous avez une docstring à respecter pour chaque fonction/procédure. Lisez les commentaires pour savoir où compléter ; en particulier, vous pouvez retirer les « return » des procédures une fois celles-ci écrites.

I. Premiers tris

1 – Coder une procédure `tri_insertion` qui trie une liste en place selon l'algorithme par insertion.

2 – Coder une procédure `tri_fusion` qui trie une liste **en place** selon l'algorithme de partition-fusion.

II. Mise en place : tests et benchmarking

3 – Écrire une fonction `générer_tab_test` qui prend en paramètre des entiers n, $mini$ et $maxi > mini$, et qui renvoie une liste de taille n d'entiers compris dans l'intervalle $[mini; maxi[$. On rappelle que `randint(a, b)` renvoie un entier aléatoire dans l'intervalle $[a; b]$.

4 – Proposer un jeu de tests pour un algorithme de tri en place, et les ajouter dans la procédure `tester_tri` prenant en paramètre une procédure de tri, partiellement fournie.

5 – Tester la validité les deux premières procédures `tri_insertion` et `tri_fusion`.