
MPSI - ITC - 7 JUIN 2026 - TP 31

RÉVISIONS : TRIS, COMPLEXITÉ, TESTS

Fonctions à effets de bord vs. fonctions pures

On dit qu'une fonction a un *effet de bord* si :

- Elle modifie les arguments qui lui sont passés en entrée
- Elle modifie des variables globales du programme
- Elle modifie le contenu de fichiers sur le disque dur
- Elle interagit avec l'utilisateur (affichage graphique ou en console)

Une fonction sans effet de bord est dite *fonction pure*.

Lien avec l'aspect *en place* d'un tri. Un tri qui est une fonction pure ne peut être en place. En effet puisqu'une fonction pure ne peut modifier l'entrée, ni modifier des variables globales, ni afficher quoi que ce soit, sa seule façon de transmettre à l'utilisateur le résultat du tri est de *renvoyer* une *nouvelle liste*, de même taille que celle en entrée, qui en est une permutation triée. Par conséquent sa complexité mémoire est linéaire en la taille de l'entrée : ce n'est pas un tri en place.

Par contre, un tri en place a forcément des effets de bords : il modifie la liste en entrée.

Conséquence sur les tests de fonctions de tri Les tris qui sont des fonctions pures peuvent être testés avec `doctest`. Par contre si un tri en place ne renvoie rien et n'affiche rien, il ne peut pas être testé avec `doctest`. Il convient alors de coder une fonction de test dédiée, qui procède en 3 étapes pour chaque valeur à tester :

- On met dans une variable la liste à trier
- On appelle la fonction de tri
- On vérifie que la liste est triée après l'appel

Exercice 1 (Tests pour tris en place et tris fonctionnels purs)

1. Sur papier, choisissez sur quelles valeurs (*liste en entrée, sortie attendue*) sur lesquelles vous voulez tester les tris que vous coderez pendant ce TP.
2. Implémentez une fonction `test_tri_en_place(tri)` qui prend en argument une fonction de tri et lève une erreur si cette fonction ne passe pas le jeu de tests choisi.
3. Implémentez une fonction `test_tri_fonctionnel_pur(tri)` qui fait la même chose mais pour une fonction de tri qui est pure.

Exercice 2 (Tri insertion)

1. Lancez une pièce où un dé pour décider si vous allez coder le tri insertion en place ou sous forme de fonction pure.
2. Implémentez le tri insertion
3. Testez-le
4. Donnez, en la justifiant, la complexité du tri insertion

Exercice 3 (Tri sélection)

1. Lancez une pièce où un dé pour décider si vous allez coder le tri sélection en place ou sous forme de fonction pure.
2. Implémentez le tri sélection
3. Testez-le
4. Donnez, en la justifiant, la complexité du tri sélection

Exercice 4 (Tri partition-fusion)

1. Implémentez le tri partition-fusion sous forme de fonction pure (sauf si vous vous appelez Olivier, dans lequel cas faites-le en place)
2. Testez votre tri
3. Donnez, en la justifiant, la complexité du tri partition-fusion

Exercice 5 (Tri à bulles)

1. Lancez une pièce où un dé pour décider si vous allez coder le tri à bulles en place ou sous forme de fonction pure.
2. Implémentez le tri à bulles
3. Testez-le
4. Donnez, en la justifiant, la complexité du tri à bulles

Exercice 6 (Tri comptage)

1. Implémentez le tri comptage sous forme de fonction pure
2. Testez votre tri
3. Donnez, en la justifiant, la complexité du tri comptage