



Exercices - Algorithmes de tris

S6 - Algorithmique (1)

Les exercices précédés du symbole  sont à faire sur machine, en sauvegardant le fichier si nécessaire.

Les exercices précédés du symbole  doivent être résolus par écrit.

! Important

Notebook Capytale pour les exercices 1 et 2 : [Capytale](#)

Exercice 1

Cet exercice est à faire dans Capytale.

Écrire une fonction `trie_chaine` qui prend en argument une **liste** de chaînes de caractères et qui **modifie** cette liste en la triant en fonction du nombre de lettres. Cette fonction ne renvoie rien.

Tester la fonction avec la liste `["un", "deux", "trois", "quatre", "cinq", "six", "sept", "huit", "neuf", "dix"]`.

Exercice 2 : le tri à bulles

Cet exercice est à faire dans Capytale.

L'algorithme de tri à bulles est le suivant :

- On parcourt la liste de gauche à droite.
- Si deux éléments consécutifs sont dans le mauvais ordre, on les échange.
- Si, à l'étape précédente, au moins un échange a eu lieu, on recommence à l'étape 1.
- Sinon, la liste est triée et on arrête.

1. Écrire toutes les étapes du tri à bulles pour la liste `[5, 3, 2, 4, 1]`.
2. Soit n un entier naturel non nul et L une liste de n entiers rangés dans l'ordre décroissant (pire des cas). Combien d'échanges sont nécessaires pour trier L dans l'ordre croissant ? En déduire une évaluation de la complexité de cet algorithme.
3. Écrire une fonction `tri_bulles` qui prend en argument une **liste** de nombres et qui **modifie** cette liste en la triant par ordre croissant en utilisant l'algorithme du tri à bulles. Cette fonction ne renvoie rien.
4. Ajouter une variable `compteur` dans la fonction `tri_bulles` qui compte le nombre d'échanges effectués. Ce nombre doit être renvoyé par la fonction. Tester la fonction avec la liste `[5, 3, 2, 4, 1]` et vérifier que le compteur vaut bien 6.

T.P. : Bilan et compléments

Important

Notebook Capytale pour ce T.P. : [Capytale](#)

Ce T.P. est à faire dans Capytale en suivant le lien ci-dessus.