



Chapitre 10

Tris TP

Simon Dauguet
simon.dauguet@gmail.com

2 mai 2025

Exercice 1 (Prise en main des tris) :

Le but de cet exercice est de s'entraîner à coder les différents tris et de les comparer.

1. Faire une fonction `motAlea(n:int) -> str` qui renvoie un mot de n lettres minuscules aléatoires. On rappelle l'existence des fonctions `chr` et `ord`.
2. Faire une fonction `motComptage(mot:str) -> str` qui prend en argument une chaîne de caractères `mot` et qui ordonne les lettres du mot par ordre alphabétique inverse (donc la lettre z en premier, puis y, puis ...).
3. Faire une fonction `motBulles(mot:str) -> str` qui fasse la même chose mais avec un tri à bulles.
4. Faire une fonction `motInsertion(mot:str) -> str` qui fasse la même chose mais avec un tri par insertion.
5. Faire une fonction `motSelection(mot:str) -> str` idem avec un tri par sélection.
6. Faire une fonction `motFusion(mot:str) -> str` avec un tri fusion.
7. Faire une fonction `comparaison() -> list` qui crée 500 mots de longueurs 500, applique chacun des fonctions précédentes sur chacun des mots, enregistre le temps d'exécution de chacun des appels des fonctions, et renvoie le temps moyens mis par chacune des fonctions.

Exercice 2 :

On considère une fichier `nombres.txt` contenant à chaque ligne un nombre généré au hasard. Proposer une procédure `triFichierNb(nomFichier: str, nomTrie: str) -> None`, qui lit le fichier `nomFichier` et trie son contenu dans le fichier `nomTrie`.

On prendre bien garde à commencer par faire une copie du fichier de travail dans son dossier personnel et de travailler avec cette copie, pour ne pas polluer la source du fichier afin de ne pas gêner ses camarades.

Exercice 3 :

On considère le fichier `classe.csv` contenant à chaque ligne, le nom de l'étudiant suivi de sa moyenne (par exemple `"Alpha,12.3\n"`). Proposer une procédure `triClasse(nomFichier: str, nomTrie: str) -> None`, qui lit le fichier `nomFichier` et trie ses lignes selon la moyenne dans le fichier `nomTrie`.