Table des matières

TP : Tri

Le but de ce TP est d'étudier deux (ou trois) méthodes pour classer par ordre croissant ou décroissant les éléments d'une liste.

I Calculs sur les listes

Exercice 1. Créer une liste L de 100 élements aléatoires.
Afficher la liste.

Exercice 2. Créer une fonction `mini` qui prend en argument une liste et retourne la valeur du plus petit élément de la liste.
Faire de même avec maxi.

Exercice 3. Créer une fonction `second_mini` qui prend en argument une liste et retourne la valeur du deuxième plus petit élément de la liste.

Exercice 4. Créer une fonction `appartient` qui prend en argument une liste et une variable `a` et retourne True si `a` est dans la liste et False sinon.

Exercice 5. Créer une fonction `combien` qui prend en argument une liste et une variable `a` et retourne le nombre de fois où a appartient à la liste si `a` est dans la liste et False sinon.

Exercice 6. Créer une fonction `combien_index` qui prend en argument une liste et une variable `a` et retourne une liste des positions des `a` de la liste si `a` appartient à la liste et False sinon

Exercice 7. Créer une fonction `supprimer_element` qui prend en argument une liste et une variable `a` et retourne une liste identique à celle prise en argument mais sans aucun `a`.

Exercice 8. Créer une fonction `anagramme` qui prend en argument une chaine de caractères et retourne un anagramme quelconque de cette chaine.

Exercice 9. Créer une fonction `nombre_anagrammes` qui prend en argument une chaine de caractères et retourne le nombre d'anagrammes de ce mot.

Exercice 10. Créer une fonction `ordre_alphabetique` qui compare deux lettres et retourne les deux lettres dans le sens de l'ordre alphabétique.

II Tri à bulles ♡

Exercice 11. Le procédé de tri à bulles consiste à mimer la séparation de liquides non miscibles de densités différentes : ainsi, les valeurs les plus grandes de la liste vont remonter au bout de la liste, comme une bulle d'huile remonterait en haut d'un verre d'eau.
Cet algorithme de tri n'est pas performant mais il est très facile à comprendre.
On se donne une liste $X = [x_1, \ldots, x_n]$ de réels.

- ⋆ Si $x_1 > x_2$, on échange les valeurs de x_1 et x_2, puis si $x_2 > x_3$, on échange les valeurs de x_2 et de x_3... et ainsi de suite jusqu'à la fin de la liste. Après cette suite d'opérations, on obtient une nouvelle liste que l'on note à nouveau $[x_1, \dots, x_n]$, où cette fois x_n est le plus grand élément de la liste.
- ⋆ On opère de même avec $[x_1, x_2, \dots, x_{n-1}]$ puis avec $[x_1, \dots, x_{n-2}]$ et ainsi de suite jusqu'à $[x_1, x_2]$.
- ⋆ Lorsque l'on a classé la liste $[x_1, x_2]$, alors la liste de départ est classée par ordre croissant.

Écrire une fonction qui trie une liste de nombres réels par ordre croissant grâce au procédé du tri à bulles.

Que faut-il modifier pour trier par ordre décroissant ?
Combien d'opérations au maximum doit-on effectuer ?

Exercice 12. ♡Écrire une fonction qui calcule la médiane d'une liste de nombres.

III Tri par insertion ♡

Exercice 13. Le tri par insertion est très similaire au tri que l'on effectue naturellement lorque l'on met en ordre un jeu de cartes : on part d'une première carte, puis on insère petit à petit les autres cartes en respectant leur ordre. La méthode est la suivante : on se donne une liste $X = [x_1, \dots, x_n]$ de réels.

- ⋆ La liste $[x_1]$ est triée.
- ⋆ On insère l'élément x_2 au bon endroit par rapport à x_1, soit avant si $x_2 < x_1$, soit après. On obtient alors une liste de deux éléments qui est bien triée.
- ⋆ On insère de la même façon x_3, puis x_4, et ainsi de suite jusqu'à x_n, pour obtenir la liste entièrement triée.

Écrire une fonction qui trie une liste de nombres réels par ordre croissant grâce au procédé du tri par insertion.

Que faut-il modifier pour trier par ordre décroissant ?
Combien d'opérations au maximum doit-on effectuer ?

IV Diviser pour mieux régner - Tri fusion

Cette partie est peu guidée et n'est pas à savoir refaire. Elle permet néanmoins de se familiariser avec les fonctions récursives.

L'idée du tri fusion est de procéder par récursivité :

- Si la liste contient qu'une seule valeur la liste est triée.
- Si il y a au moins deux valeurs, couper le tableau en deux, trier (récursivement) les listes obtenues, puis fusionner les résultats. La fusion consiste à rassembler dans une seule liste trié les valeurs contenues dans les tableaux triés fournis en paramétre .

On va tout d'abord proposer un algorithme pour traiter la fusion de deux listes `L1, L2`. Ecrire une fonction python `fusion` qui suit la procédure suivante :

- Si l'une des listes est vide, renvoyer l'autre
- Sinon renvoyer la liste formée par la plus petite des deux valeurs `L1[0]` et `L2[0]`, suivie de la fusion des listes où l'on a enlevé la valeur que l'on vient de placer (eg. `L1[1:]`)

Ecrire une fonction Python `tri` qui permet de programmer l'algorithme de tri par la méthode de fusion proposé en début de paragraphe. (On pourra finir cette fonction par `return(fusion(tri(L[:m]),tri(L[m:]))))` où m correspond à la moitié de la taille de la liste)