

TP 9 – Tris.

Exercice 1 – Tri à bulles Pour trier une liste selon la méthode du tri à bulles, on réalise des balayages successifs : à chaque balayage, on compare deux éléments consécutifs du tableau et on les réordonne.

Cette méthode doit son nom au fait qu'il déplace rapidement les plus grandes valeurs à la fin du tableau, comme une bulle d'air qui remonte à la surface d'un liquide.

Question 1 En utilisant la liste $[10, 3, 7, 5, 9, 7, 8, 0, 8]$, écrire la séquence d'échanges permettant d'arriver à la liste triée.

Question 2 Écrire une fonction `triBullesNaif(L : list) -> None` qui prend en argument une liste L et qui trie cette liste par une méthode naïve selon la méthode du tri à bulles.

Question 3 Pour quel type de tableau, est-on dans le meilleur des cas ? Quelle est alors la complexité de l'algorithme ?

Question 4 Pour quel type de tableau, est-on dans le pire des cas ? Quelle est alors la complexité de l'algorithme ?

Question 5 En remarquant qu'à l'étape i , les i derniers éléments sont triés, proposer une nouvelle fonction `triBulles(L : list) -> None` nouvel algorithme du tri à bulles.

Question 6 En remarquant qu'à l'itération i , il est possible d'arrêter le tri là où la dernière inversion a eu lieu à l'étape $i - 1$, proposer une nouvelle fonction `triBullesOptimise(L : list) -> None` du tri à bulles.

Exercice 2 – Tri par comptage d'une liste

On suppose que la liste à trier L est constituée d'entiers de l'intervalle $\llbracket 0; k \rrbracket$. L'algorithme fonctionne suivant le principe suivant. On parcourt une fois la liste et on compte le nombre d'éléments de la liste égaux à $0, 1, \dots, k - 1$. Pour ce faire on utilise une liste C de taille k . On peut alors facilement procéder à une réécriture de la liste initiale, de sorte qu'en sortie elle soit constituée des mêmes éléments, mais triés dans l'ordre croissant.

Ainsi, si on cherche à trier la liste $L = [2, 1, 4, 1]$. Dans un premier temps on crée une liste C contenant cinq fois la valeur 0 : $C = [0, 0, 0, 0, 0]$. Une fois le comptage terminé on obtient la liste suivante : $C = [0, 2, 1, 0, 1]$. La liste triée sera donc constituée de 2 fois la valeur 1 puis 1 fois la valeur 2 puis une fois la valeur 4.

L'algorithme prend en entrée la liste L à trier, ainsi qu'un entier k tel que tous les éléments de la liste soient des entiers de l'intervalle $\llbracket 0; k \rrbracket$. On procède en deux étapes : d'abord compter les éléments de chaque type, ensuite réécrire la liste L .

Question 1 Écrire la fonction on pourra (au choix) utiliser l'une des signatures suivante : `tri_comptage(L: list, k: int) -> None` ou `tri_comptage(L: list, k: int) -> list` permettant de réaliser un tri par comptage (avec ou sans effet de bord).

Question 2 La fonction proposée agit-elle avec effet de bord ? Sans effet de bord ?

Question 3 La fonction proposée réalise-t-elle un tri stable ? un tri en place ?

Exercice 3 – Tris d'une base de données des films de cinéma

Objectif

Réaliser un tri numérique et un tri alphabétique à partir d'une base de données

On donne le fichier `films_martiniere_2018.csv` dans lequel un peu plus de 2000 films sont référencés avec le titre, l'année de création, le réalisateur et le box office.

Une proposition de lecture du fichier csv et de création de la liste de films est donnée ci-dessous et dans le fichier `lecture_fichier_csv.py` :

```
f=open('films_martiniere.csv','r')
ligne=f.readline()
fichier=f.readlines()
f.close()
L=[]
for ligne in fichier:
    ligne=ligne.replace('"','')
    ligne=ligne.split(';')
    ligne[-1]=ligne[-1].rstrip('\n')
    ligne[-1]=int(ligne[-1])
    ligne[1]=int(ligne[1])
    L.append(ligne)
```

Avec le logiciel Pyzo, vous devez avoir votre dossier visible dans la fenêtre **file browser** ou alors exécuter le fichier en cliquant droit sur l'onglet de votre nom de fichier et sélectionner *Exécuter en tant que script*. Votre fichier python et le fichier `films_martiniere.csv` doivent être dans le même dossier.

Question 1 Commenter chaque ligne du fichier `lecture_fichier_csv.py`. Copier vos tests dans le script python.

Question 2 Choisir l'algorithme de tri le plus efficace pour trier le fichier de 2000 films (autre que `sort`). Copier l'algorithme choisi dans votre script et modifier-le afin qu'il puisse trier une liste de listes.

Question 3 Trier les films en fonction du box office. Quel est le film qui a été le plus vu au cinéma ?

Question 4 Définir la fonction `comparer(L:list)` qui a pour argument une liste `L` de deux mots `mot1:str` et `mot2:str` et qui renvoie cette liste triée par ordre alphabétique. Les mots de la liste seront écrits en lettres majuscules. Les titres de films peuvent comporter des chiffres.

Question 5 Implémenter un algorithme de tri alphabétique adapté au fichier `films_martiniere_2018.csv`. Quel est le titre du premier film de la liste ?