TD 2

Travaux dirigés

Patricia Bessonnat & Xavier Pessoles

Savoirs et compétences :

- ❒ *Alg – C17 : tris d'un tableau à une dimension de valeurs numériques (tri par insertion, tri rapide, tri fusion).*

Exercice 1 – Représentation du coût temporel des tris

Objectif Représenter pour chacun des tris les courbes indiquant le temps d'exécution en fonction du nombre d'éléments à trier.

On donne la bibliothèque de tri `tris.py` dans laquelle différents tris ont été implémentés. On dispose ainsi des fonctions :

- `tri_insertion`;
- `tri_rapide`;
- `tri_fusion`.

On dispose aussi de la méthode `sort` disponible en Python.

On utilisera de plus le module `time` de la bibliothèque `time` pour créer un chronomètre et le module `randint` de la bibliothèque `random`.

Pour augmenter la limite de récursivité de Python, on utilisera les instructions suivantes : `import sys` puis `sys.setrecursionlimit(100000)`.

Question 1 *Tracer, dans chacun des 4 cas, le temps de tri d'une liste en fonction du nombre d'éléments de la liste. Le nombre d'éléments variera de* 0 *à* 1 000 *par pas de 100. Une liste de n éléments sera composée de nombres choisis aléatoirement entre 0 et n. Ce réseau de courbes représentera le cas moyen.*

Question 2 *Tracer, dans le cas des trois tris les plus rapides, le temps de tri d'une liste en fonction du nombre d'éléments de la liste. Le nombre d'éléments variera de* 0 *à* 20 000. *Une liste de n éléments sera composée de nombres choisis aléatoirement entre 0 et n. Ce réseau de courbes représentera le cas moyen.*

Question 3 *Conclure sur l'efficacité algorithme de chacun des tris dans le cas moyen.*

Question 4 *Comparer les temps de tris de chacune des 4 méthodes sur une liste triée d'un million d'éléments. Que peut-on en conclure ?*

Exercice 2 – Classement de l'étape Tarbes – La Pierre-Saint-Martin – 167 km

Les coureurs du tour de France sont en train de terminer la seizième étape du Tour de France qui sépare Tarbes et La Pierre-Saint-Martin.

Le fichier `classement_général` rassemble le classement général à l'issue de l'étape 9. Le fichier `etape_10` contient le classement de l'étape 10 uniquement. Dans le fichier texte, les champs sont séparés par des tabulations.

Objectif L'objectif est de réaliser le classement général après la seizième étape.

Lecture des fichiers de résultat

Question 1 *Réaliser la fonction* `charge_classement` *permettant de lire un fichier de classement et de retourner une liste de la forme* `[[Nom_1, Dossard_1, Temps_1], [Nom_2, Dossard_2, Temps_2], ...]`. *Le temps devra être exprimé en secondes.*

Classement en fin d'étape

Dans une première approche, on souhaite réaliser le classement général après la fin de l'étape.

Question 2 *Réaliser la fonction permettant d'ajouter les temps de l'étape 10 aux temps du classement général.*

Question 3 *Quel méthode de tri vous semble la mieux adaptée au tri du classement général ?*

Question 4 *Modifier les algorithmes de tris pour pouvoir trier la liste donnée suivant le temps de course d'un coureur. Le classement général a-t-il changé à l'issue de la dixième étape ?*

R Travaillant sur une liste de listes, la méthode `sort` n'est plus adaptée. On peut donc utiliser la fonction, `sorted` en utilisant une clef de tri (la clef correspondant à la colonne sur laquelle on souhaite trier la liste) :

```
# Tri de la liste <<liste>> sur la colonne 3
sorted(liste, key=lambda colonnes: colonnes[2])
```

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.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)
```

Pour ceux qui travaillent 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 programme proposé de lecture du fichier csv. Copier vos tests dans le script python.*

Question 2 *A partir de l'étude réalisée dans l'exercice 2, choisir l'algorithme de tri le plus efficace pour trier le fichier de 2000 films. 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` *qui a pour argument une liste de deux mots* `mot1` *et* `mot2` *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 de films. Quel est le titre du premier film de la liste ?*