TD – 02

Exercices d'applications

Savoirs et compétences :

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

On donne la bibliothèque de tri `tris.py` dans laquelle différents tris ont été implémentés (même fichier que pour le TD01). On dispose ainsi des fonctions :

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

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

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

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

Les coureurs du tour de France sont en train de terminer la dixième 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 dixième 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])
```

Classement en cours d'étape – Implémentation d'une file

On cherche à reconstituer le classement général au fur et à mesure que les coureurs arrivent. Dans cette partie le classement de l'étape (liste de listes) sera vu comme une **file** FIFO (First In First Out) où le premier élément est le premier coureur arrivé et le dernier élément est le dernier coureur à avoir passé la ligne d'arrivée.

Question 5 *Implémenter les fonctions élémentaires liées à la gestion des files :* `enfiler`, `defiler`, `est_vide`. *À l'intérieur de ces fonctions, on s'autorise les méthodes liées aux listes (*`append`, `pop`, *...).*

Question 6 *Implémenter la fonction* `ajout` *ayant pour but d'ajouter le temps de l'étape d'un coureur dans le classement général et de mettre à jour ce classement. La gestion du classement de l'étape devra être réalisé grâce à une liste.*

Question 7 *Implémenter la fonction* `ajout` *ayant pour but d'ajouter le temps de l'étape d'un coureur dans le classement général et de mettre à jour ce classement. La gestion du classement de l'étape devra être réalisée grâce à une liste.*

Question 8 *Quelle pourrait être l'utilité de la fonction* `enfiler` *dans un tel contexte ?*