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).*

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`.

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 2 000 000. Une liste de n éléments sera composée de nombres choisis aléatoirement entre 0 et n. Ce réseau de courbes représentera le cas moyen.*

Question 2 *Conclure sur l'efficacité algorithme de chacun des tris dans le cas moyen.*

Exercice 2 – Classement de l'étape Bourg-de-Péage – Gap

Les coureurs du tour de France sont en train de terminer la seizième étape du Tour de France qui séparait Bourg-de-Péage et Gap d'une distance de 201 km.

Le fichier `classement_général` rassemble le classement général à l'issue de l'étape 15. Le fichier `etape_16` contient le classement de l'étape 16 uniquement.

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

Préambule

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 16 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 seizième étape ?*

Classement en cours d'étape – solution intuitive

On cherche à reconstituer le classement général au fur et à mesure que les coureurs arrivent.

Question 5 *Implémenter la fonction* `ajout` *ayant pour but d'intégrer le temps de l'étape d'un coureur et de mettre à jour le classement.*

Classement en cours d'étape – arbre binaire

Un tas binaire est une structure de données informatiques qui permet d'accéder au maximum (respectivement minimum) d'un ensemble de données en temps constant. On peut la représenter par un arbre binaire vérifiant deux contraintes :

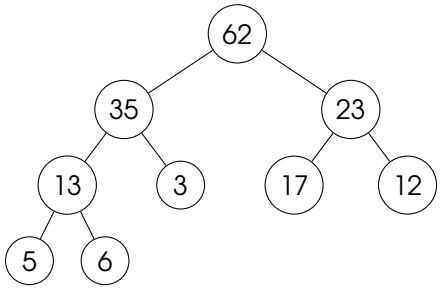
- c'est un arbre binaire parfait : tous les niveaux de l'arbre (excepté le niveau le plus bas) sont totalement remplis. Si le dernier n'est pas totalement rempli alors il doit l'être de gauche à droite ;
- c'est un tas : une clé est associée à chaque nœud de l'arbre. Cette dernière doit être supérieure ou égale aux clés de chacun de ses fils.

Ainsi, lorsque les clés sont des nombres (valeurs) et quand la relation d'ordre choisie est l'ordre naturel, on parle alors de tas-max (ou max-heap).

Un tas binaire étant un arbre binaire complet, on peut l'implémenter à l'aide d'un tableau tel que :

- la racine de l'arbre (niveau le plus haut) se trouve à l'index absolu 1 ;
- en considérant un nœud à l'index absolu i :
  - son fils gauche se trouve à l'index absolu $2i$;
  - son fils droit se trouve à l'index absolu $2i+1$;
- en considérant un nœud à l'index absolu $i > 1$:
  - son père se trouve à l'index absolu $i/2$, le symbole / désignant ici la division entière.

La figure suivante illustre un arbre binaire (trié) et le tableau qu'on peut lui associer avec les index associés (index absolu, index en langage Python).



Valeur dans l'arbre (clé)	62	35	23	13	3	17	12	5	6
Index absolu dans l'arbre	1	2	3	4	5	6	7	8	9

Le nœud ayant la valeur 62 est la racine de l'arbre (niveau le plus haut). Les nœuds ayant les clefs 5 et 6 sont au niveau le plus bas.