Info3 A, TP 6

Exercice 1: (*listes*)
Si vous n'avez pas fait cet exercice, il faut le faire.

Programmer en Python la class liste représentée comme un Tuplet (tete, queue). Attention, il ne s'agit pas de la classe liste vue en TD avec les attribues data et next. Inclure les fonctions usuelles de manipulation de listes (longueur, générer la liste contenant les entiers entre deux entiers i et j donnés en paramètres, afficher, renverser, append pour concaténer deux listes, pairs pour rendre les éléments de rang pais, fusion de deux listes, tri fusion, tri rapide, appliquer une fonction donnée sur tous les éléments de la liste, …)
Application : une fonction pour créer la liste représentant un entier écrit dans une base donnée, une fonction pour construire un nombre à partir de la liste de ses chiffres.

Exercice 2: *(Arbres)*

- Quelles sont les différentes façons de traverser d'un arbre binaire de recherche (BST= Binary Search Tree),
- Montrer que la traversée d'un BST donne une séquence ordonnée des éléments constituant l'arbre,
- Programmer en Python la structure de données arbre binaire de recherche et monter comment l'utiliser pour le tri d'un tableau d'entiers.

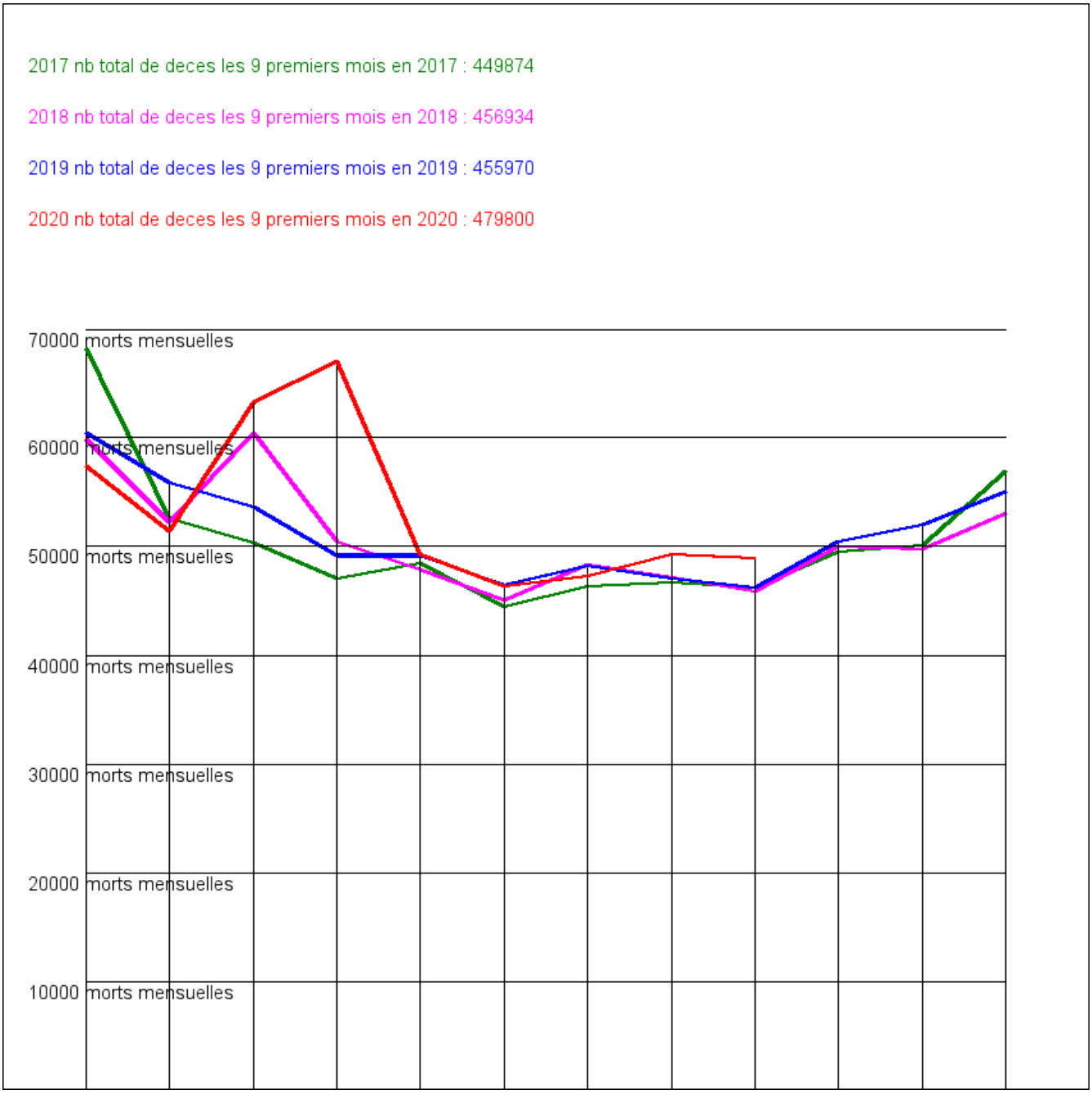
Exercice 3 : (Covid-19)

L'objectif de ce TP est de générer le dessin ci-dessous à partir des donnée ISEE collectées sur Internet.
Chercher avec un moteur de recherche "mortalité mensuelle France insee" donne la page:
https://www.insee.fr/fr/statistiques/serie/001641603
de laquelle vous pouvez extraire les données suivantes:

Janvier=1
Fevrier=2
Mars=3
Avril=4
Mai=5
Juin=6
Juillet=7
Aout=8
Septembre=9
Octobre=10
Novembre=11
Decembre=12

insee=[
(2020, Septembre, 48900),
(2020, Aout , 49200),
(2020, Juillet , 47200),
(2020, Juin , 46300),
(2020, Mai , 49200),
(2020, Avril, 67000),
(2020, Mars, 63200),
(2020, Fevrier , 51400),

(2020, Janvier , 57400),
(2019, Decembre, 54958),
(2019, Novembre, 51905),
(2019, Octobre, 50410),
(2019, Septembre, 46181),
(2019, Aout , 47056),
(2019, Juillet , 48128),
(2019, Juin , 46468),
(2019, Mai , 49100),
(2019, Avril , 49160),
(2019, Mars, 53630),
(2019, Fevrier, 55837),
(2019, Janvier, 60410),
(2018, Decembre,52985),
(2018, Novembre,49736),
(2018, Octobre,49993),
(2018, Septembre,45803),
(2018, Aout, 47176),
(2018, Juillet,48331),
(2018, Juin, 45027),
(2018, Mai, 47841),
(2018, Avril, 50416),
(2018, Mars, 60391),
(2018, Fevrier,52175),
(2018, Janvier, 59774),
(2017, Decembre,56937),
(2017, Novembre,50011),
(2017, Octobre,49452),
(2017, Septembre,46144),
(2017, Aout, 46640),
(2017, Juillet,46349),
(2017, Juin, 44393),
(2017, Mai, 48389),
(2017, Avril, 47025),
(2017, Mars, 50251),
(2017, Fevrier,52538),
(2017, Janvier,68145)
]



Les données arrondies sur un multiple de 100 sont temporaires, mais elles n'évolueront pas de façon statistiquement significative.

A partir de insee, construisez une table (un dictionnaire) telle que : table[(an, mois)] donne le nombre de morts correspondants, avec an entre 2017 et 2010, et mois entre 1 et 12.

Ecrire nb_deces(table, an, mois) qui rend le nombre de décès cet an et ce mois, ou -1 si l'information n'est pas dans la table.

Ecrire compter(table, mois1, mois2) qui rend un dictionnaire total tel que total[an] donne le nombre total de décès cet an entre le mois1 (typiquement 1) et le mois2 (9 pour Septembre).

Pour dessiner une arête (un trait) ou un texte, il vous faudra transformer vos coordonnées x et y dans les

coordonnées de l'image (entre 0 et width). Votre coordonnée x est un mois (entre 1 pour janvier et 12 pour décembre), votre coordonnée y est un nombre de morts mensuels : au plus 70 mille. Attention, l'axe des y est descendant. Centraliser la transformation dans une fonction (appelée par exemple dans les procédures edge() et texte() qui tracent une arête et un texte), et appeler toujours edge() et texte(). Vous pouvez écrire une fonction d'interpolation linéaire telle que :

v=interpolation((u1, v1), (u2, v2), u)

retourne v égal à v1 quand u égale u1, retourne v égal à v2 quand u égale u2. Les u sont dans les coordonnées du monde (mois, ou nombre de décès) ; les v sont dans les coordonnées de l'image. Testez d'abord cette fonction d'interpolation avant de vous lancer dans le graphique.

Ecrire le programme qui génère une image similaire à celle donnée en haut.

```
#en tête :
import os
from PIL import Image, ImageDraw, ImageFont


#taille de l'image, en nombre de pixels, largeur=hauteur
width=850

#pour créer un canevas width par width :
    img=Image.new("RGB", (width,width), (255,255,255))
    c = ImageDraw.Draw(img)        # c pour canvas
    # après avoir tracé dans le canevas :
    img.show()                  #pour afficher l'image
    img.save("dessin.png")  #pour la sauver au format png
    img.save("dessin.pdf")   #pour la sauver au format pdf
    img.save("dessin.ps")     #pour la sauver au format postscript

#pour tracer un texte en x, y. c est le « canevas » :
def texte( c, x, y, chaine, couleur):
    font=ImageFont.truetype( "/usr/share/fonts/truetype/msttcorefonts/arial.ttf", 15)
       # remplacer par le « bon fichier »...
    c.text( (x, y), chaine, fill=couleur, font=font)

#pour tracer une arête :
def edge( c, x1, y1, x2, y2, couleur, epaisseur):
    c.line( ( x1, y1, x2, y2), fill=couleur, width=epaisseur)
```

Python fournit de nombreuses bibliothèques graphiques, malheureusement incompatibles entre elles. PIL ne donne pas de beaux résultats graphiques mais est simple d'emploi.

Exercice 4: *(Projet)*
Commencer à réfléchir sur le projet, e.g. programmer la somme et le produit de deux polynômes.