Tp ALGORITHME des k plus proches voisins

L'objectif de ce travail est d'implémenter un algorithme qui permette à partir d'une liste échantillon de mots français et allemands, de pouvoir prédire si un mot mystère quelconque est plus de consonnance française ou allemande.

1. Travail préparatoire sur papier

Pour réaliser cette prédiction, on décide ici de tenir compte uniquement de la longueur du mot mystère et du nombre de voyelles qui le composent. On souhaite prédire l'origine des mots suivants : « anoure », « ohrwurm » et « vitrine ».

1- Compléter les tableaux suivants :

Echantillon :

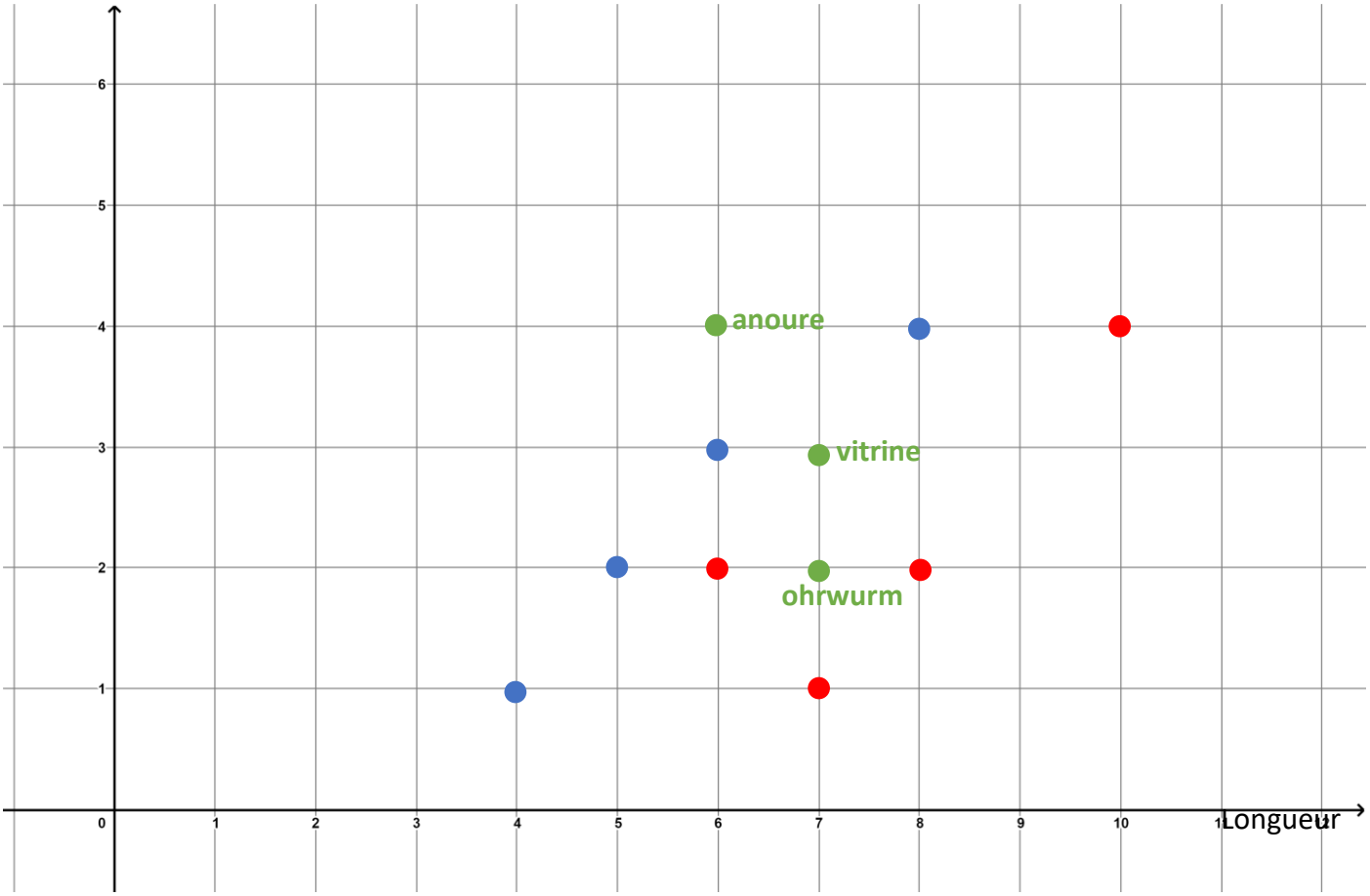
Mots	*Longueur*	*Nombre de voyelles*	*Origine*
hiver	5	2	français
loup	4	1	français
reines	6	3	français
nordique	8	4	français
winter	6	2	allemand
schloss	7	1	allemand
koniginnen	10	4	allemand
nordisch	8	2	allemand

Mots mystères dont on souhaite déterminer l'origine :

Mots	*Longueur*	*Nombre de voyelles*
anoure	6	4
ohrwurm	7	2
vitrine	7	3

2- Représenter les données de l'échantillon et des mots mystères sur le graphique qui suit et qui représente le nombre de voyelles des mots en fonction de la longueur : utiliser des points bleus pour les mots français, rouge pour les allemands et verts pour ceux dont on souhaite déterminer l'origine.

Nb de voyelles



3- En utilisant des arguments graphiques, donner si possible, une prédiction quant à l'origine des mots « anoure », « ohrwurm » et « vitrine » :

En fonction de l'emplacement des points sur le graphe, on peut classer les mots-mystères :

Mots	***Origine***
anoure	français
ohrwurm	allemand
vitrine	français

2. Implémentation d'un algorithme de prédiction de la langue d'origine d'un mot

Le code donné sur la page suivante permet à l'exécution d'obtenir :

```
>>> (executing file "langue_voisins.py")
nombre de voisins : 3
fr fr al
```

ou :

```
>>> (executing file "langue_voisins.py")
nombre de voisins : 5
fr fr al fr fr
```

On peut en déduire que le mot « *anoure* » est d'origine française. Ce résultat est en accord avec l'étude sur papier réalisée dans la partie

```python
from math import sqrt

def nb_voyelle(mot) :
    '''
        Argument : mot est une chaine de caractère (string)
        Renvoie le nombre de voyelle de mot (int)
    '''
    voyelles = "aeiouy"
    n = 0
    for c in mot :
        if c in voyelles : n += 1
    return n

def distance(mot,mot_mystere):
    '''
        Arguments :
            mot (string) et mot_mystere (string)
        Renvoie la distance (float) entre un mot et un mot mystère : distance = sqrt((len()-len())²-(nb_voy()-nb_voy())²
    '''
    delta_l = len(mot) - len(mot_mystere)
    delta_v = nb_voyelle(mot) - nb_voyelle(mot_mystere)
    d = sqrt(delta_l**2 + delta_v**2)
    return d

def tri_selection(liste,mot_mystere,k) :
    '''
       Arguments :
            liste : liste dont les éléments sont des listes du type [mot (string) , langue (string)]
            mot_mystere (string) : mot dont on veut connaître l'origine
            k (int) : nombre de voisins souhaités
        Modifie l'ordre de liste, on mettant en début de liste et dans l'ordre, les k éléments pour lesquels la distance
        avec le mot_mystere est la plus petite. On utilise un algo du type "tri par sélection".
    '''
    if k >= len(liste)-1 : k = len(liste)-1
    for i in range(k) :
        jMin = i
        for j in range(i+1,len(liste)-1) :
            if distance(liste[j][0],mot_mystere) < distance(liste[jMin][0],mot_mystere) :
                jMin = j
        liste[i],liste[jMin] = liste[jMin],liste[i]
    return k

def kProchesVoisins(echantillon , mot_mystere) :
    '''
        Arguments :
            echantillon : liste dont les éléments sont des listes du type [mot (string) , langue (string)]
            mot_mystere (string) : mot dont on veut connaître l'origine
        Ce code écrit dans le shell la langue d'origine des k plus proches voisins
    '''
    k = int(input('nombre de voisins : '))
    k = tri_selection(echantillon,mot_mystere,k)
    for i in range(k) :
        print(echantillon[i][1] , end=" ")

# Main
petit_echantillon = [
    ["hiver","fr"],
    ["loup","fr"],
    ["reines","fr"],
    ["nordique","fr"],
    ["winter","al"],
    ["schloss","al"],
    ["koniginnen","al"],
    ["nordisch","al"]
]
kProchesVoisins(petit_echantillon , "anoure")
```

On demande d'écrire et de compléter ce code dans une fichier nommé « *kVoisinsLangue.py* », dans l'ordre suivant :

1- Ecrire la liste « *petit_echantillon* »
2- Ecriture et test de la fonction *nb_voyelle()*
3- Ecriture et test de la fonction *distance ()*
4- Ecriture et test de la fonction *tri_selection()*
5- Ecriture et test de la fonction *kProchesVoisins()*

6- Conclusion de ce Tp