Lycée Voltaire

Mini Projet

Graphes

On dispose d'une variable DICO qui est une liste de mots d'un même nombre de lettres.
Voici un exemple avec des mots de quatre lettres (cf dico.py) :

DICO = [”aime”, ”cime”, ”dure”, ”gris”, ”lame”, ”mars”, ”orge”, ”part”, ”purs”, ”saie”, ”tari”, ”auge”, ”cire”, ”durs”, ”haie”, ”lime”, ”mere”, ”ours”, ”paru”, ”rage”, ”sale”, ”tige”, ”baie”, ”cris”, ”fart”, ”hale”, ”lire”, ”mers”, ”page”, ”pere”, ”raie”, ”sape”, ”toge”, ”brie”, ”cure”, ”fors”, ”hors”, ”loge”, ”mime”, ”paie”, ”pers”, ”rale”, ”sari”, ”tore”, ”bris”, ”dame”, ”gage”, ”hure”, ”luge”, ”mire”, ”pale”, ”pipe”, ”rame”, ”scie”, ”tors”, ”bure”, ”dime”, ”gaie”, ”iris”, ”mage”, ”mors”, ”pame”, ”pire”, ”rape”, ”sure”, ”tort”, ”cage”, ”dire”, ”gais”, ”juge”, ”maie”, ”muet”, ”pane”, ”pore”, ”rare”, ”taie”, ”trie”, ”cale”, ”ducs”, ”gale”, ”jure”, ”male”, ”mure”, ”pape”, ”prie”, ”rime”, ”tale”, ”tris”, ”came”, ”dues”, ”gare”, ”kart”, ”mare”, ”murs”, ”pare”, ”pris”, ”rire”, ”tape”, ”troc”, ”cape”, ”duos”, ”gars”, ”laie”, ”mari”, ”nage”, ”pari”, ”pues”, ”sage”, ”tare”, ”truc”]

Le problème que l'on se pose est le suivant : on se donne deux mots **mot1** et **mot2** de DICO et on cherche à trouver, si il existe, une suite de mots de DICO telle que :

- la suite commence par **mot1** et se termine par **mot2**
- deux mots consécutifs de la suite ne diffèrent que d'une lettre, sans tenir compte de l'ordre des lettres dans chacun des mots (on dira qu'ils sont voisins).

Par exemple avec le mot ”truc” et le mot ”aime”.
Une suite possible est : 'truc' − > 'cure' − > 'cire' − > 'cime' − > 'aime'
Il s'agit donc de trouver une telle suite des mots permettant d'aller d'un mot de DICO à un autre en passant uniquement par des mots voisins.

Exercice 1 *Compléter la fonction* ***SontVoisins*** *ci-dessous, qui renvoie True si les deux mots sont voisins (s'ils différent d'une seule lettre) et False sinon. (Voir les exemples ci-dessous)*

```
def SontVoisins(mot1,mot2) :
    """
    param mot1, mot2 : (str) same size
    return result : (bool)
    >>>SontVoisins("bonjour", "onsbouj")
    True
    >>>SontVoisins("bonjour", "ojurate")
    False
    >>>SontVoisins("bonjour", "jourboa")
    True
    >>>SontVoisins("bonjour", "bonbonr")
    False
    """
    if len(mot1) != len(mot2) :
        raise Exception("Les mots doivent avoir la même taille")

    ...

    return result
```

Exercice 2 *1. Compléter la fonction* ***ListeVoisins*** *qui renvoie la liste des mots voisins d'un mot donné en paramètre.*

```
def ListeVoisins(mot) :
    """
    param mot : (str) a word
    return result : (list) liste des mots qui sont voisins.
    """
    result = []
    ...
    ...
    ...
    return result
```

2. *Quels sont les voisins du mot "ours" ?*

Exercice 3

1. *Appliquer un parcours en profondeur pour aller du mot "ours" au mot "cage".*
2. *Appliquer un parcours en largeur pour aller du mot "ours" au mot "cage".*