
ANAGRAMMES

UE4 Base de données avancées - Mini-projet

EXERCICE 1

Le script est constitué de 2 parties.

Dans la 1ère partie, on range tous les mots du dictionnaires dans une liste `stocker`. Dans une cette liste, on prends on choisi un mot que l'on range dans une liste `K` et on cherche ses anagrammes que l'on range dans une liste `V`. Puis, on prends une nouvelle liste `newT` qui est le complémentaire de `stocker` et les mots sélectionnés. La liste `newT` sera utilisé pour choisir à nouveau les anagrammes. On applique ceci pour chaque mot. À la fin, on obtient deux listes : `K` où se trouve des mots et `V` les anagrammes de ces mots.

Dans la 2e partie, on range dans un *data frame* avec le mot témoin en 1ère colonne et ses anagrammes en 2e colonne. Dans un second *data frame*, on présente le nombre d'ensemble ayant *i* anagramme.

L'algorithme prends 1mn pour traiter seulement 0,67% du dictionnaire. Pour traiter la totalité du dictionnaire, l'algorithme prends 2h48.

On conclut que l'algorithme est très lent et qu'il est nécessaire de l'améliorer.

EXERCICE 2

Pour améliorer la vitesse d'exécution, la première solution était d'utiliser plus fréquemment des fonctions Python.

1er : Pour chaque mot du dictionnaire inclue dans la liste `stocker`, on tri chaque lettre dans l'ordre alphabétique puis on le range dans une liste `B`. Pour chaque mot dans `B`, on crée une liste booléenne appelée `C`. La liste `C` contient des doublons de liste, on créer une liste `D` qui écarte tous les doublons. Dans une nouvelle liste `E`, on trouve l'index des `True` pour chaque liste dans `D`. Enfin, grâce à la liste `E`, on révèle les mots à partir des index pour chaque liste que l'on range dans la liste `F`. La vitesse de ce script est encore très lent.

Une autre piste pour améliorer la vitesse d'exécution, réduire l'utilisation de boucle et la création de liste; penser à utiliser les dictionnaires et recourir aux fonctions Python.

2e : Pour chaque mot se trouvant dans une liste `stocker`, on tri chaque lettre dans l'ordre alphabétique puis on range dans une liste `B`. Ces mots sont rangés dans un dictionnaires `di co` en tant que clé et ont comme valeur une liste vide. Avec la fonction `enumerate`, on range les anagrammes dans la liste correspondant proprement au mot.

La vitesse de ce script est très rapide : seulement 0,51 s.