

Projet de TP module Algorithmique : Partie I

Partie I.- Dictionnaire de fréquences des mots et ses applications

Dans les études du langage nous nous intéressons à l'étude des contextes dans lesquels un mot est employé. Pour un mot m , son contexte d'ordre k est défini par la suite des k mots qui le suivent.

Exemple

Soit T une chaîne de caractères :

$T = \text{"Le langage C est un langage de programmation impératif généraliste."}$

— Contexte d'ordre 1 du mot "langage" :

$$\text{Contexte}(\text{"langage"}, 1) = \{C, de\}$$

— Contexte d'ordre 2 du mot "langage" :

$$\text{Contexte}(\text{"langage"}, 2) = \{(C, est), (de, programmation)\}$$

Remarque : Dans le reste de l'exercice on se limite au contexte d'ordre 1.

1. Définir le type **listeMots** qui représente une liste de chaînes de caractères.
2. Écrire une fonction **listeMots motsDe(char* t)** qui retourne la liste des mots d'un texte t . Notons que les caractères lexicographiques sont aussi considérés comme des mots.
3. Une liste de fréquences est une liste chaînée qui associe à chaque mot m du texte t sa fréquence d'apparition.
 - (a) Définir un type **ListFreq** pour déclarer cette liste.
 - (b) Écrire une fonction **listeFreq frequenciesDe(char* t)** qui retourne la liste, triée en ordre croissant, des fréquences des mots de t .
4. Soient t un texte et m un mot. Écrire la fonction **listeFreq contexte(char* T, char* m)** qui retourne la liste des contextes (d'ordre 1) de m (triée par ordre décroissant des fréquences).
5. Un dictionnaire de contextes d'un texte t , stocke pour chaque mot $m \in T$ la liste de ses contextes (d'ordre 1). Définir un type **dictContext** pour représenter ce dictionnaire et écrire la fonction qui le crée à partir d'un texte donné.
6. Étant donné le dictionnaire des contextes et un mot m , écrire la fonction **char* maxContexte(dictContext d, char* m)** qui retourne le mot le plus fréquent dans le contexte de m .
7. Écrire la fonction **int freqCont(dictContext d, char* m, char* n)** qui retourne le nombre d'apparition du mot n dans le contexte de m .

8. Écrire la fonction **void genererText(dictContext d, int n)** qui affiche la séquence de n mots où le mot à la position i est le contexte le plus fréquent du mot à la position $i - 1$. Le mot à la position 1 est choisi aléatoirement à partir des mots qui figurent dans le dictionnaire.

Quelques rappels

1. Un dictionnaire est une structure de données qui permet de représenter un ensemble fini de mots et de rechercher, en temps rapide, si un mot appartient ou non à l'ensemble. Si le dictionnaire est dynamique, il doit permettre de plus d'ajouter et/ou de supprimer un mot.

Le TP doit être réalisé en binôme ou en trinôme (ni plus ni moins).

La date de remise du TP est : **18/12/2021**

BON TRAVAIL