

NF16
TP-4
PENNORS JOSSELIN
TARDIO IKER

1) Partie sur les ABR :

Pour la définition de la structure noeud de l'ABR, on a décidé de créer une structure composé d'une chaîne de caractère et de deux pointeurs vers les noeuds gauche et droit de notre noeud.

Notre type DicoABR représente un pointeur vers la structure noeud défini précédemment.

DicoABR Creernoeudvide() :

Cette fonction permet de créer un nouveau noeud où toutes ces variables sont initialisées à 0. Elle renvoie le dictionnaire mis à jour avec l'ajout de ce nouveau noeud.

DicoABR initdico () :

Cette fonction initialise un nouveau dictionnaire avec un appel à la fonction creernoeudvide().

DicoABR creernoeud (char* newmot) :

Cette fonction crée un noeud initialisé avec la chaîne de caractère newmot entrée en paramètre, puis ces noeuds gauche et droit initialisés par la fonction creernoeudvide.

DicoABR ajoutMot (DicoABR dico, char* newmot) :

La fonction ajoutMot parcourt l'arbre de recherche avant d'insérer un mot, on se retrouve donc en $O(h)$.

DicoABR rechercheMot (DicoABR dico, char* newmot) :

La fonction rechercheMot parcourt l'arbre de recherche, on est donc en $O(h)$.

DicoABR predecesseur (DicoABR source) :

Cette fonction recherche le prédecesseur de notre mot en récupérant le maximum du sous arbre gauche de notre noeud.

DicoABR supprimeMot (DicoABR dico, char* newmot) :

La fonction supprimeMot parcourt l'arbre grâce à la fonction rechercheMot, puis le supprime, on est donc comme avec la fonction rechercheMot en $O(h)$.

int comparaison (char* mot1, char* mot2, char* souschaine) :

Cette fonction nous permet de comparer deux mots avec une souschaine et renvoie le mot qui à le plus de similitude avec la souschaine.

void insertion (int k, DicoABR dico, char* souschaine, char resultats[][20]) :

Cette fonction permet d'insérer dans un tableau les mots que l'on à comparé avec la fonction comparaison.

void suggestionsMotsParcours (int k , DicoABR dico, char* souschaine, char resultat[][20]) :

Cette fonction parcourt l'intégralité de notre dictionnaire en utilisant la fonction insertion pour réaliser un tableau contenant les mots qui correspondent le plus à notre chaine.

void suggestionsMots (int k, DicoABR dico, char* souschaine, char resultats [k][20] :

Cette fonctions permet de faire un appel à à fonction suggestionsMotsParcours pour l'ensemble de notre tableau pour avoir un tableau complet de mots qui ont le plus de similitude avec notre chaine.

DicoABR chargerDicoABR (DicoABR dico, char* adresse) :

Cette fonctions initialise un dictionnaire avec la fonction initDico(), puis ouvre un fichier et lie sur ce fichier les mots ligne par ligne et ajoute chaque mot à notre dictionnaire grace à la fonction ajoutMot(dico, mot).

void afficherABR (DicoABR dico) :

Cette fonction permet d'afficher notre dictionnaire dans l'ordre alphabetique.

int contient (char* mot, char* souschaine) :

Cette fonction permet d'afficher les mots de notre dictionnaire qui contiennent une souschaine de notre mot rechercher.

void afficherABRchaine (DicoABR dico, char* souschaine) :

Cette fonction traverse l'ensemble de notre dictionnaire et affiche les mots qui contiennent une partie de notre mot souschaine.

void printABR (DicoABR dico) :

Cette fonction permet d'afficher les mots du dico ou bien d'afficher les mots qui contiennent une sous chaîne.

void verimotABR (DicoABR dico, char* adressedico, char* adressefichier) :

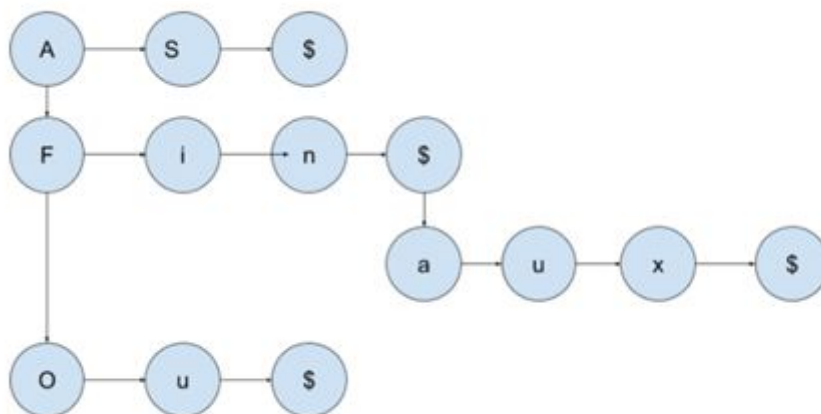
Cette fonction permet de parcourir notre ABR et pour chaque mot propose des suggestions et demande à l'utilisateur s'il desire ajouter ce mot, ou remplacer ce mot par l'une des suggestions.

void veridicoABR (DicoABR dico, char* adressedico, char* adressefichier) :

Cette fonction nous permet, à partir de la recherche d'un mot, de changer ce mot dans le dictionnaire à partir de différentes suggestions, de supprimer ce mot du dictionnaire ou alors de le valider.

2) Partie sur les arbres lexicographiques

Voici l'exemple du dictionnaire une fois la suppression du mot final.



Le programme est fait de nombreuses fonctions. Au lancement de l'arbre lexicographique le fichier dictionnaire.txt est chargé. Il y a ainsi une large base de mots.

L'utilisateur a accès aux fonctions :

1/ Initialisation d'un dictionnaire.

Cette fonction est à utiliser uniquement si on désire réinitialiser le dictionnaire et en créer un nouveau. Tous les mots seront alors perdus.

2/ Chercher le plus long préfixe.

Soit un préfixe entré par l'utilisateur. La fonction renverra le plus long préfixe du mot. Exemple s'il y a les mots fin, final et finalement dans la base, la fonction renverra « finalement » si on entre les caractères « finalement ».

3/ Affichage des éléments.

Cette partie se décompose en 3 parties. Il est possible d'afficher le dictionnaire en entier, d'afficher les mots commençant par le préfixe souhaité, d'afficher les mots contenant le nombre de caractères souhaité.

4/ Chercher un mot dans le dictionnaire

Il faut dans un premier temps entrer un mot dans le terminal. Puis le programme va parcourir le dictionnaire et chercher ce mot. Il renverra un entier indiquant s'il a été trouvé ou non.

5/ Ajouter un nouveau mot au dictionnaire

Tout nouveau mot peut être ajouté au dictionnaire. La fonction va insérer selon les modalités désirées comme présentées sur l'image au-dessus.

6/ Supprimer un mot du dictionnaire.

Tout mot présent dans le dictionnaire peut être supprimé. Ainsi le dictionnaire retourné ne le contiendra plus et les modifications nécessaires auront été apportées.

7/ Suggestion de mots

Cette fonction rencontre certaines difficultés qui se répercutent ainsi sur les options 8 et 9. Elle permet de sélectionner les mots les proches d'un mot lambda entré. Soit les mots « final », « fin », « fine » et « fins » entrés. Si on entre le mot « finalement » la fonction renverra dans l'ordre décroissant « final », « fins », « fine » et « fin ».

8/ Vérimot

La fonction ouvre un fichier « file.txt » et y extrait tous les mots présents. Pour chaque mot elle propose 3 cas :

- Remplacer le mot du fichier par un autre. L'utilisateur peut entrer le mot de son choix et le modifier.
- Ajouter le mot sélectionné dans le dictionnaire
- Ne rien faire en faisant « Retour ».

Tout au long du programme les mots seront réécrits dans un fichier « test2.txt ». Cela permettra en cas de remplacement du mot de le changer aisément. Ce fichier sera renommé en « file.txt » à la fin.

9/ Véridico

Après que l'utilisateur ait entré un mot, une suggestion d'au maximum 5 mots lui sera retournée. La fonction permet de choisir entre trois options une fois le mot souhaité sélectionné. Le mot peut être corrigé et ainsi modifié. Il peut être supprimé. Une dernière option est de le valider ou de faire retour, ce qui correspond à la même chose.

10/ Quitter.

Cette dernière fonction aura pour effet de lancer la fonction dechargerDicoAI. Cette dernière va parcourir l'intégralité du dictionnaire et copier tous les mots un à un dans un fichier test.txt. Ce dernier sera une fois la fonction terminée renommé « dictionnaire.txt ». Ainsi le dictionnaire sera mis à jour.