

## 4TPM205U: Algorithmique des tableaux: feuille 6

### Recherche dichotomique dans tableau trié

À traiter après la feuille 7 en 2019-2020

## Travaux dirigés sur machine

Depuis la page <http://dept-info.labri.fr/ENSEIGNEMENT/algotab/src/> téléchargez les fichiers `tp06.py` et `dictionnaireSansAccents.txt`.

Vous ajouterez au fichier `tp06.py` le code demandé ci-dessous.

### Exercice 1

L'objectif de cet exercice est de vérifier si un mot donné fait partie d'un dictionnaire de mots français. On manipulera donc des tableaux de chaînes de caractères.

Le fichier `tp06.py` contient le code nécessaire pour construire le tableau `t_mots` de chaînes de caractères, contenant tous les mots du fichier texte `dictionnaireSansAccents.txt`. Ainsi, chaque case du tableau contiendra une chaîne de caractères correspondant à un mot du dictionnaire. En Python, on peut tester si deux chaînes de caractères sont dans l'ordre lexicographique avec l'opérateur `<`. Un bémol : ça ne fonctionne que si les chaînes contiennent uniquement des caractères alphabétiques minuscules non accentués. On peut également tester si deux chaînes de caractères sont identiques avec l'opérateur `==`.

1. Combien de mots contient le dictionnaire ?
2. Écrire une fonction qui vérifie si un tableau est trié dans l'ordre croissant et l'appeler pour vérifier que le tableau de chaînes de caractères `t_mots` est bien trié dans l'ordre alphabétique.
3. Écrire une fonction `rechercheBalayage(t, n, ch)` qui renvoie l'indice dans le tableau `t` de la chaîne de caractères `ch` si elle s'y trouve et -1 sinon. Dans cette première version de la recherche d'une chaîne de caractère, on parcourt séquentiellement tout le tableau. Votre code serait-il différent si on traitait un tableau de nombres entiers ?
4. Est-ce que *zymotique* est un mot du dictionnaire utilisé ? Et *yaourth* ? Combien de temps prend la fonction `rechercheBalayage` pour rechercher chacun de ces mots dans le tableau `t_mots` ?<sup>1</sup>
5. Écrire une fonction `rechercheDichotomique(t, n, ch)` qui renvoie l'indice dans le tableau `t` de la chaîne de caractères `ch` si elle s'y trouve et -1 sinon. Comme son nom l'indique cette fonction implémente l'algorithme de recherche dichotomique.
6. Vérifiez que la fonction `rechercheDichotomique` donne les résultats attendus sur les mots *zymotique* et *yaourth*. Combien de temps prend-elle pour rechercher chacun de ces mots dans le tableau `t_mots` ?

---

1. Pour calculer le temps d'exécution en millisecondes, on appellera la fonction `tempsExecution` fournie dans le fichier `tp06.py` en lui donnant comme arguments la fonction de recherche, le tableau `t_mots`, sa taille et le mot recherché.

## Exercice complémentaire 1

Reprenez les questions de l'exercice précédent mais en utilisant le dictionnaire `dictionnaire.txt` disponible sur la page <http://dept-info.labri.fr/ENSEIGNEMENT/algotab/src/>. Ce dictionnaire contient des mots accentués et d'autres contenant des tirets.

Pour information les caractères accentués présents dans ce dictionnaire sont :

à, â, ä, ç, é, è, ê, ë, î, ï, ô, ö, ù, û, ü
---

Pour comprendre quel problème se pose, vous pourrez chercher le mot *troussequinâmes* qui existe bien dans le dictionnaire.

Est-il trouvé avec la fonction `rechercheBalayage` ?

Et avec la fonction `rechercheDichotomique` ?

Modifiez votre code de façon à ce que toutes les versions de la recherche d'une chaîne de caractère fonctionnent sur le dictionnaire avec accents.

Effacer les fichiers `dictionnaireSansAccents.txt` et `dictionnaire.txt` qui consomme beaucoup de place.