

1 Le mot le plus long

1.1 Principe

Le but de ce projet est de réaliser un solveur pour le jeu “le mot le plus long”. Dans ce jeu, 10 lettres sont choisies aléatoirement, et les joueurs doivent réaliser un mot le plus long possible en utilisant au plus une fois chacune de ces 10 lettres.

Exemples :

tirage	meilleurs solutions
eeetlbn	lente
iaofnfd	fondai
auurnng	gruau nauru
ieevclc	cilice civile
aainsrv	navrais ravinas savarin
euasrlt	luteras talures
coirdss	dossier
ooemclt	ocelot
eielnns	enlies enlise ensile sienne

1.2 Consignes pour démarrer le TP

Nous vous fournissons :

- un fichier dictionnaire `dict.txt`, contenant une liste de mots de la langue française, au format UTF8, un mot par ligne, sans autre formatage.
- un autre fichier dictionnaire `dict-simple.txt`, contenant la même liste de mot, mais sans accent.
- une liste de tirages `tirages.txt`, contenant sur chaque ligne une liste d’au plus 10 lettres parmi les 26 de l’alphabet, en majuscule.

Tout cela est dans le dépôt `git` initial, disponible [ici](#).

1.3 Tâches à accomplir

1.3.1 Tâche 1

Le programme doit écrire dans un fichier, pour chaque tirage, le ou les plus longs mots du dictionnaire `dict.txt` pouvant être formés avec ce tirage. Le dictionnaire simplifié sans accent peut être utilisé tant que votre programme ne gère pas les accents correctement. Les mots admissibles peuvent en effet comporter des accents, l’algorithme doit ignorer les accents.

1.3.2 Tâche 2

le programme offre un option permettant de jouer, en affichant une liste de lettres tirées aléatoirement, en demandant la réponse de l’utilisateur, et en affichant la liste des meilleures réponses possibles. Les nombres de consonnes et de voyelles doivent être choisis par l’utilisateur, et la probabilité des lettres doit être conforme à leur usage en français (le 's' est plus fréquent que le 'w').

1.3.3 Consignes

C'est à vous de décider quelles classes écrire et quelles méthodes doit avoir chaque classe. Voici quelques indications pour vous aider.

- Commencer par des objectifs plus simples. Par exemple, au début utilisez une tirage fixe directement encodé dans votre programme. De même utilisez un dictionnaire fixe, avec peu de mots et peu de lettres distinctes.
- Chaque tâche est découparable en sous-tâches, certaines étant communes, la plupart peuvent être implémentées indépendamment les unes des autres. Faites-les une par une. Tester !
- Utiliser des algorithmes simples. S'ils ne sont pas assez performants, prévoyez de pouvoir les changer plus tard par des algorithmes plus complexes.
- La liste des tirages est longue, le dictionnaire aussi. Si vos algorithmes ne sont pas très efficaces, votre programme n'y arrivera pas. Ce n'est pas grave ! Générez un fichier de tirages plus petit avec la commande `head`.
- Il existe un algorithme efficace pour trouver le mot le plus long, utilisant un arbre préfixe. Attention, c'est difficile, assurez-vous que vous avez déjà un algorithme naïf qui fonctionne avant de vous lancer dans ce travail. Ce sera cependant nécessaire si vous voulez résoudre tous les tirages en un temps raisonnable.
- Utiliser le dictionnaire sans accent. Supprimer les mots contenant des caractères non-alphabétiques. Si vous avez tout fait, renseignez-vous sur comment faire pour supprimer les accents d'un mot, afin de pouvoir utiliser un dictionnaire avec accent (ce qui permet d'utiliser n'importe quelle dictionnaire).
- Vous trouverez facilement les fréquences d'usage des lettres en langue française en faisant une petite recherche. Si vous avez du temps, considérez réaliser un calcul à partir du dictionnaire fourni.