

Implanter les algorithmes de recherche exacte d'un mot dans un texte suivants :

- algorithme naïf, avec boucle interne, sans boucle rapide, sans sentinelle ;
- algorithme naïf, avec boucle interne, avec boucle rapide, sans sentinelle ;
- algorithme naïf, avec boucle interne, avec boucle rapide, avec sentinelle ;
- algorithme naïf, avec `strncmp`, sans boucle rapide, sans sentinelle ;
- algorithme naïf, avec `strncmp`, avec boucle rapide, sans sentinelle ;
- algorithme naïf, avec `strncmp`, avec boucle rapide, avec sentinelle ;
- algorithme de Morris-Pratt ;
- algorithme de Knuth-Morris-Pratt ;
- algorithme de Boyer-Moore ;
- algorithme de Horspool ;
- algorithme Quick Search.

Écrire un premier générateur pseudo-aléatoire de texte permettant de stocker dans des fichiers des textes d'une longueur donnée sur un alphabet de taille donnée.

Utiliser ce générateur pour produire des textes de longueur 500 000 sur des alphabets de taille 2, 4, 20 et 70.

Écrire un second générateur pseudo-aléatoire de mots permettant de stocker dans des fichiers un nombre de mots donné, d'une longueur donnée sur un alphabet de taille donnée.

Utiliser ce générateur pour produire des ensembles de 100 mots de longueur 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40 et 50 sur des alphabets de taille 2, 4, 20 et 70 (écrire un script `bash`).

Pour chacun des textes générés effectuer la recherche des ensembles de mots générés sur le même alphabet.

Il est donc bien évident qu'il faut créer deux générateurs indépendamment des algorithmes de recherche.

Relever les temps d'exécution de chacune de ces recherches, faire des courbes et les commenter en les comparant aux complexités théoriques des différents algorithmes.

Donner les meilleurs algorithmes en fonction de la longueur des mots à rechercher et de la taille de l'alphabet.