

1 Problématique

Rechercher les occurrences d'un mot dans un texte est une fonctionnalité intégrée dans tous les logiciels de traitements de texte. Si la tâche semble aisée dans un texte d'une seule page, nous pouvons nous interroger sur l'efficacité de la recherche d'un mot dans un livre entier.

Comment effectuer une recherche textuelle efficace ?

2 Approche naïve

La première méthode à laquelle nous pouvons penser consiste à :

- observer une **fenêtre** du texte,
- dans cette fenêtre, comparer chaque lettre du **motif** recherché au texte,
- décaler la fenêtre d'un cran dès qu'il n'y a pas de correspondance.

	0	1	2	3	4	5	6	7	8	9	10
texte	a	d	r	a	c	a	d	a	b	r	a
motif	d	a	b								
	0	1	2								

FIGURE 1 – Première comparaison : pas de correspondance

	0	1	2	3	4	5	6	7	8	9	10
texte	a	d	r	a	c	a	d	a	b	r	a
motif	d	a	b								
	0	1	2								

FIGURE 2 – Première comparaison : correspondance

	0	1	2	3	4	5	6	7	8	9	10
texte	a	d	r	a	c	a	d	a	b	r	a
motif	d	a	b								
	0	1	2								

FIGURE 3 – Deuxième comparaison : pas de correspondance

2.1 Implémentation

Activité 1 :

1. Écrire la fonction `recherche_naive(texte: str, motif: str) → int` qui renvoie la position du *motif* dans le *texte* ou -1 s'il n'est pas présent.
2. Estimer la complexité temporelle de cet algorithme dans le pire des cas : le motif n'est pas présent dans le texte.

3 Approche plus efficace : Boyer-Moore

3.1 Recherche à l'envers

La première idée de cet algorithme est de commencer la recherche **en partant de la fin du motif**.

	0	1	2	3	4	5	6	7	8	9	10
texte	a	d	r	a	c	a	d	a	b	r	a
motif	d	a	b								
	0	1	2								

FIGURE 4 – Première comparaison : pas de correspondance

Pour l'instant cette approche ne semble pas apporter d'amélioration par rapport à l'algorithme précédent.

3.2 Décalages par sauts

3.3 Prétraitement du motif