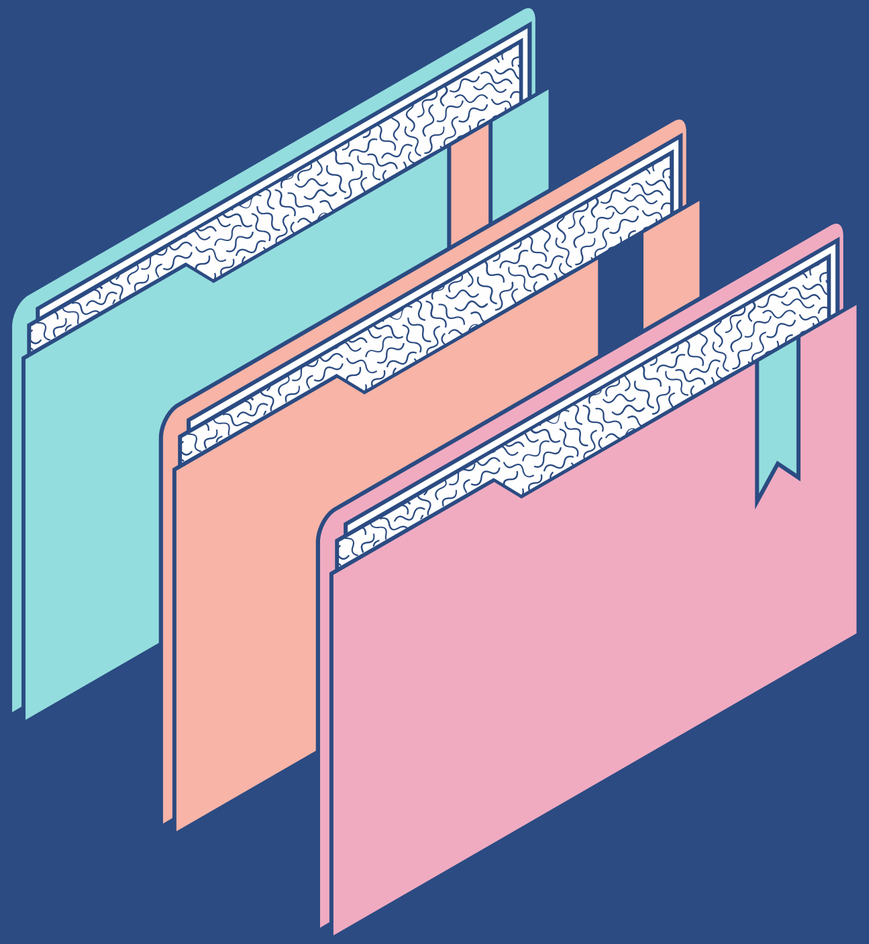
BY ISAAC, MATHIEU, THEAULT

PROJET 2 RECHERCHE TEXTUELLE



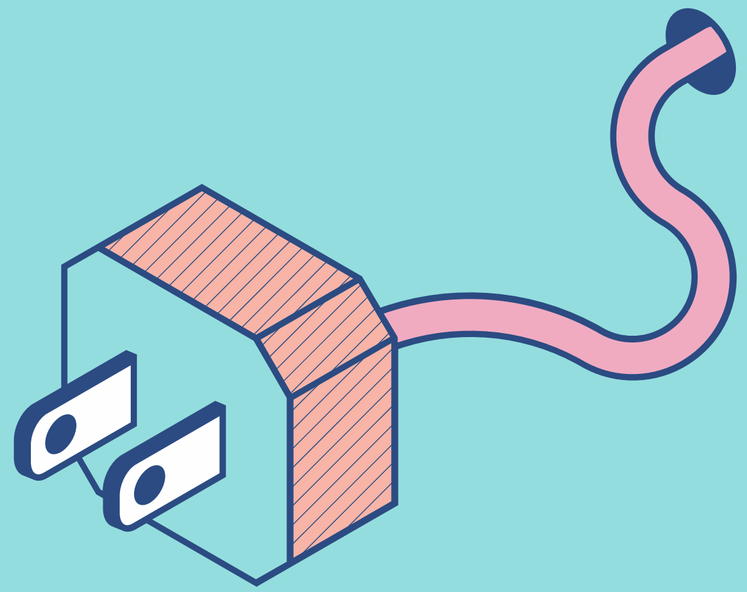
Sommaire

COMMENT VA SE PRÉSENTER LE PROJET

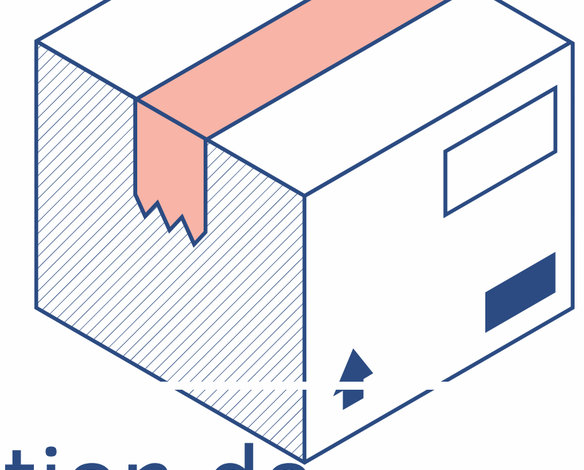
- Qu'est-ce que la recherche textuelle.
- Présentation de la partie 1, explications.
- Comment va-t-on coder ?
- Présentation de la partie 2, explications.
- Présentation de notre sujet.
- Comment va-t-on faire ?
- Notre répartition des tâches.

Qu'est-ce que la recherche textuelle ?

La recherche textuelle est une recherche de motif (bout de texte) à l'intérieur d'un texte (SQL, TXT, etc.), La recherche textuelle permet de trouver un motif dans un texte donné, cela est pratique par exemple pour les généticiens, pour chercher un gène X dans un génome d'ADN.



Dans la partie 1 il nous faudra coder 2 programme



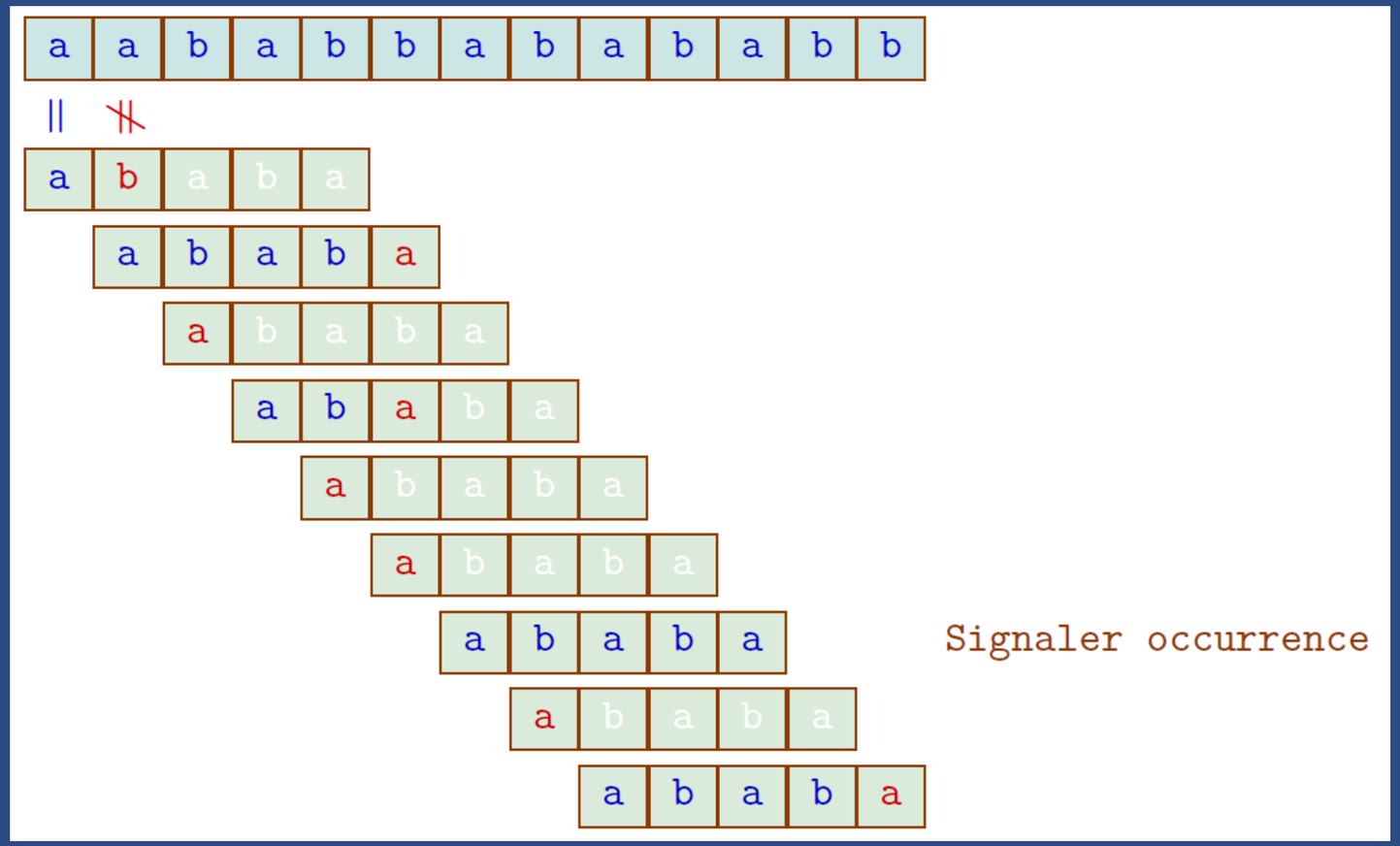
Algorithme de recherche dit "naïf" :

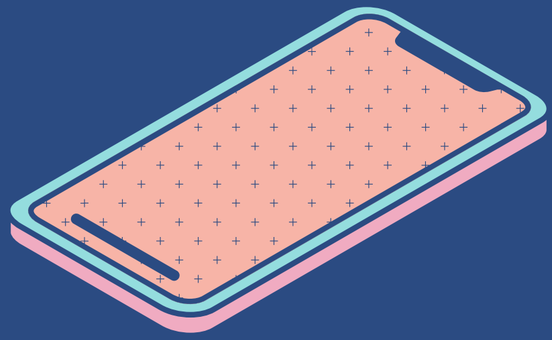
- Compare un à un.
- Compare de gauche à droite.
- En cas de non-correspondance, on avance simplement la fenêtre d’un indice vers la droite (si c'est possible).

Implémentation de l'algorithme de Boyer-Moore :

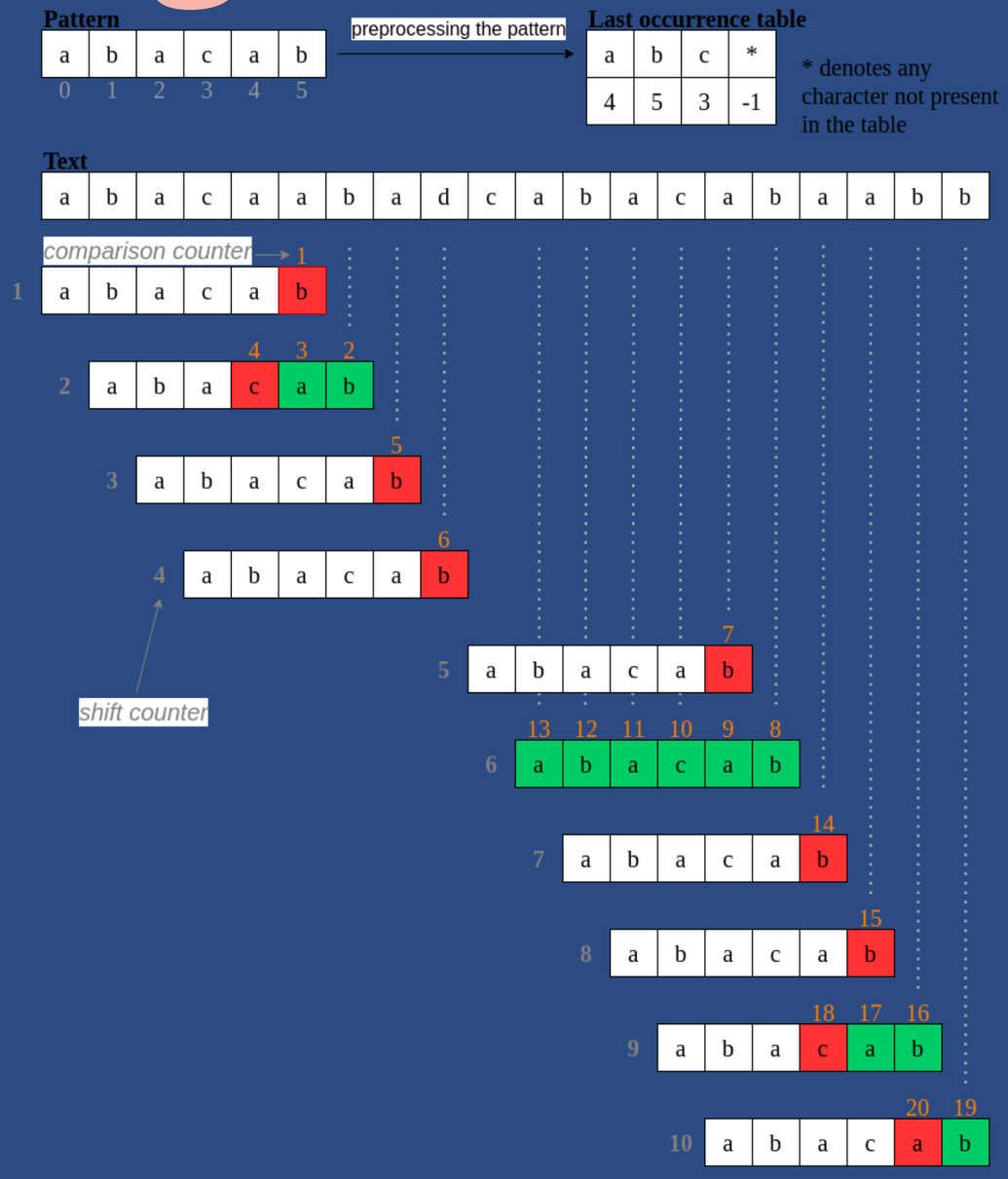
- Dans la fenêtre glissante, compare de droite à gauche.
- Applique la règle du mauvais caractère (si le caractère étudié est présent à un autre endroit dans le texte, on va cadrer la fenêtre avec ce caractère, puis on va tester) et celle du bon suffixe (qui permet de faire un plus grand saut, si le motif a des répétitions (ex : DEFDEF), et qu’on retrouve ce même DEF dans le texte).

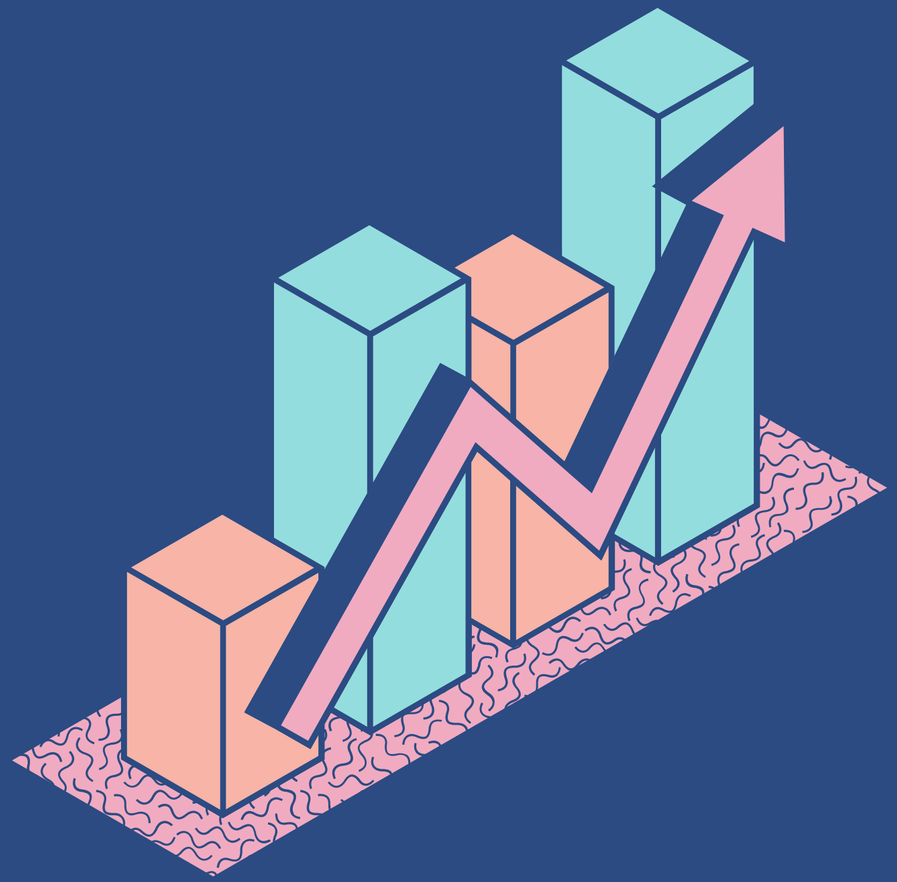
Algorithme de recherche dit "naïf"





L'algorithme de Boyer-Moore





Comment va-t-on coder cela ?

recherche naive

-Créer une fonction qui prend le texte et le motif en entrée. Initialisez les variables nécessaires comme la longueur du texte et du motif. Utiliser deux boucles imbriquées pour parcourir le texte et le motif. Comparer les caractères du motif avec ceux du texte à chaque position. Lorsqu'une correspondance est trouvée, ajouter la position à la liste des occurrences.

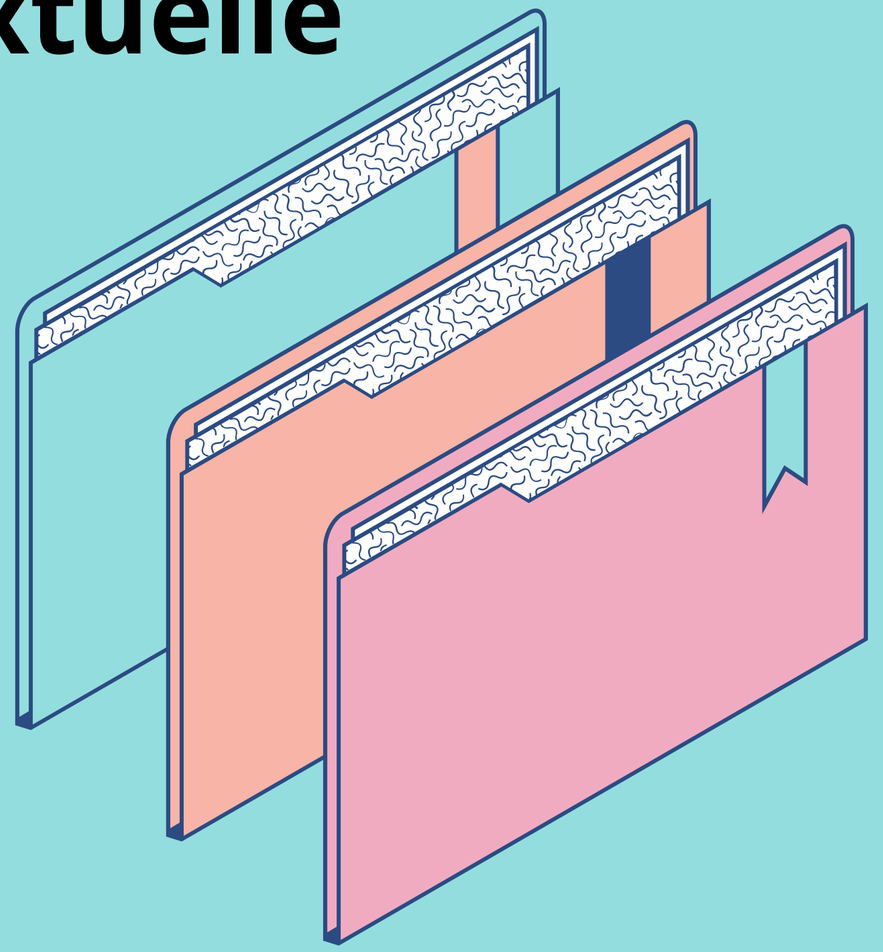
recherche par implémentation Boyer moore

– Créer une fonction qui prend le motif en entrée et retourne un dictionnaire des sauts basés sur le dernier caractère.

-Créer une fonction qui prend le texte et le motif en entrée. Initialiser les variables nécessaires comme la longueur du texte et du motif, et précalculer les sauts basés sur le dernier caractère. Utiliser une boucle pour parcourir le texte et comparer les caractères du motif en partant de la fin. Lorsqu'une non-correspondance est détectée, utiliser les deux heuristiques pour effectuer des sauts.

Dans cette partie 2 du projet il nous faudra introduire un sujet sur lequel on va appliquer notre recherche textuelle

Il faut donc :
-que ce soit utile (que notre recherche sert à quelque chose, ex : chercher une combinaison précise dans un génome d'ADN).
-que la recherche textuelle fonctionne dessus.
-eh bien évidemment faire un sujet qu'on aime😊.



Notre sujet: une notice de jeux vidéo

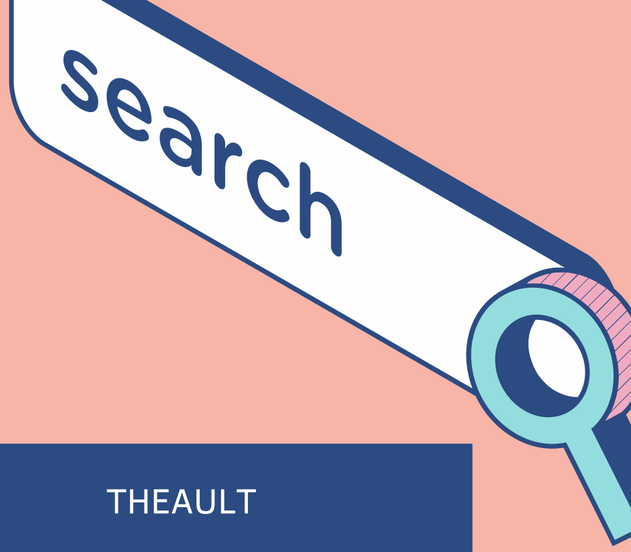
L'intérêt ?
-rechercher des informations importantes dans la notice (ex: comment sauvegarder).
-un format assez facilement manipulable disponible (.pdf).

Comment va-t-on utiliser la recherche textuelle sur notre sujet ?

Pour utiliser notre recherche textuelle sur notre sujet, nous allons utiliser PYmuPDF(ne fonctionne qu'en python 3.8 minimum!!!) (c'est une bibliothèque python a importer pour manier les fichiers pdf)sur python, il nous faudra :

- extraire le texte du fichier pdf grâce a PYmuPDF avec certaines informations (page, ligne).
- utiliser un de nos algorithme textuelle sur le texte extrait pour trouver ce que l'on cherche.
- renvoyer la page et la ligne correspondant au texte recherché pour que l'utilisateur trouve ce qu'il cherche, ou alors informer l'utilisateur si le texte n'est pas présent.

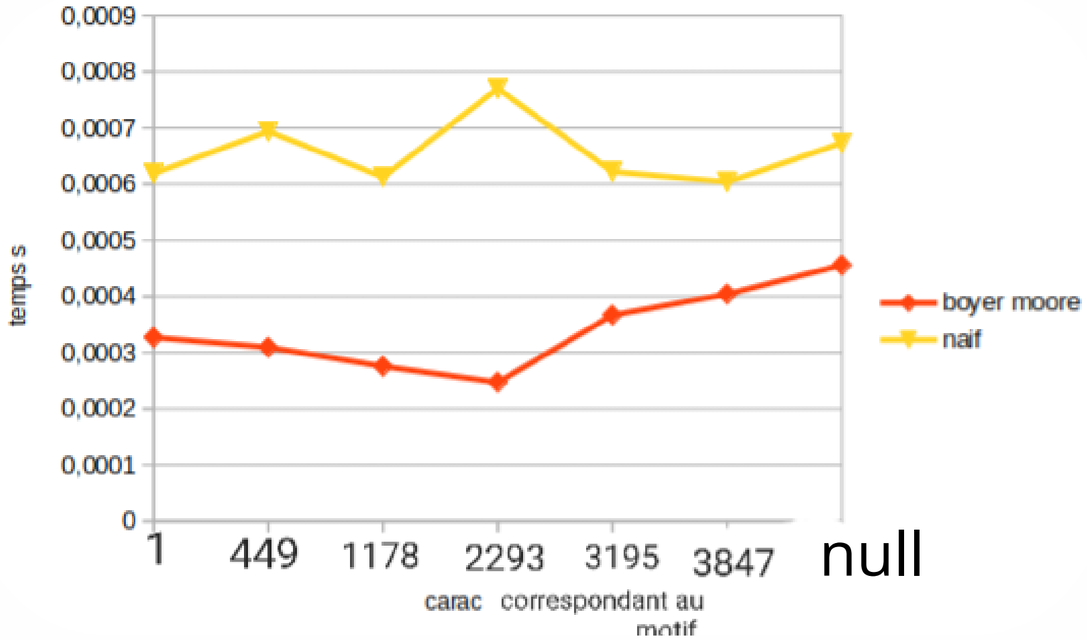
Répartion des tâches



NOTRE PROGRAMME

PROJET	MATHIEU	ISAAC	THEAULT
Semaine 1	Découverte du projet	Découverte du projet	Découverte du projet
Semaine 2	Découverte et début de la programmation de l'algorithme Boyer moore	Découverte et début de la programmation de l'algorithme naïf	Création du rapport+compréhension des algorithmes
Semaine 3	Programmation de l'algorithme Boyer moore	Programmation de l'algorithme naïf+aide sur le Boyer Moore	Rapport(explication des programmes)+recherche d'un sujet
Semaine 4	Finalisation de l'algorithme de Boyer moore (raté, on utilisera Horspool)	Finalisation de l'algorithme de Boyer moore (raté, on utilisera Horspool)	Recherche d'une bibliothèque python pour pdf puis découverte de PYpdf2
Semaine 5	Maitrise de PYmuPDF(changement de bibliothèque)	Maitrise de PYmuPDF(changement de bibliothèque)	Rapport(répartion des tâches+ début section sujet)
Semaine 6	Finalisation du projet, utilisation de PYmuPDF sur le sujet	Finalisation du projet, programmation des tests	Finalisation du rapport(fin section sujet) et aide pour l'application de PYmuPDF sur notre sujet

Benchmark



Les deux programmes sont très rapides, moins de 1 ms.

On remarquera néanmoins que le Boyer Moore dure ≈ deux fois moins de temps que le naïf en moyenne, cela nous rappelle donc qu'il est beaucoup plus rapide que le naïf, non pas par sa complexité, mais par ses heuristiques.

À l’aide du module Time, on calcule le temps d’exécution de la fonction Boyer Moore, puis on compare avec le temps d’exécution de la fonction naïve

- Il est donc normal d'utiliser Boyer Moore sur notre sujet, car il est plus performant.

Merci d'avoir lu ce rapport

Isaac, Mathieu, Théault