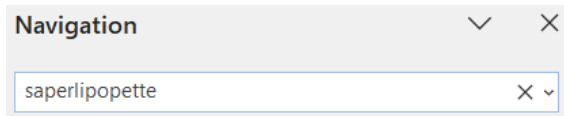


Chapitre 21 - Recherche textuelle

Dans la plupart des éditeurs de texte, il existe une fonctionnalité *Recherche* qui permet de rechercher la présence d'un motif dans un document.

Pour rechercher par exemple la présence du motif « *saperlipopette* » dans le texte ci-contre, on lance un « Ctrl F » et on saisi dans la fenêtre de dialogue :



Lorem ipsum dolor sit amet consectetur adipisicing elit. Beatae libero nesciunt illo consequatur doloribus adipisci, animi corporis molestiae ipsa perferendis ullam voluptate expedita magnam explicabo sint iste aut cumque vitae. Lorem ipsum dolor sit amet consectetur adipisicing elit. Beatae libero nesciunt illo consequatur doloribus adipisci, animi corporis molestiae ipsa perferendis ullam voluptate expedita magnam explicabo sint iste aut cumque vitae. Lorem ipsum dolor sit amet consectetur adipisicing elit. Beatae libero nesciunt illo consequatur doloribus adipisci, animi corporis molestiae ipsa perferendis ullam voluptate expedita saperlipopette magnam explicabo sint iste aut cumque vitae.

On découvre dans ce chapitre un algorithme développé par [Robert S. Boyer](#) et [J Strother Moore](#) en 1977. Il permet de réaliser une recherche avec une complexité faible.

1- ALGORITHME DE BOYER-MOORE :

Exemple : `texte = "GCATCGCAGAGAGTATACAGTACG"`
`motif = "GCAGAGAG"`

index <i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
texte	G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
Motif	G	C	A	G	A	G	A	G																
index <i>k</i>	0	1	2	3	4	5	6	7																

Exemple d'exécution avec le motif de l'exemple :

```
def dico_lettres(motif):
    dico = {}
    for i in range(len(motif)-1):
        dico[motif[i]] = i
    return dico
```

```
>>> dico_lettres("GCAGAGAG")
{'G': 5, 'C': 1, 'A': 6}
```

G	C	A	G	A	G	A	G
0	1	2	3	4	5	6	7

```

def BM(texte, motif):
    '''
    renvoie la liste des indices (éventuellement vide)
    des occurrences de la chaîne motif dans la chaîne texte.
    '''
    dico = dico_lettres(motif)
    indices = []
    i = 0
    while i <= len(texte) - len(motif):
        k = len(motif)-1
        while k >= 0 and texte[i+k] == motif[k]:
            k = k - 1
        if k == -1:
            indices.append(i)
            decalage = len(motif)
        else:
            if texte[i+k] in dico and dico[texte[i+k]] < k :
                decalage = k - dico[texte[i+k]]
            else:
                decalage = k + 1
            i = i + decalage
    return indices

```

```

>>> dico_lettres("GCAGAGAG")
{'G': 5, 'C': 1, 'A': 6}
len(motif)-1 =

```

index <i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
texte	G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
Motif	decalage = k - dico[texte[i+k]]																							
index <i>k</i>																								G
	0	1	2	3	4	5	6	7																

index <i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
texte	G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
Motif	decalage = k - dico[texte[i+k]]																							
index <i>k</i>																								G
	0	1	2	3	4	5	6	7																
Motif																								
index <i>k</i>																								G
	0	1	2	3	4	5	6	7																

⇒ Travail à faire :

- Ecrire le code des fonctions dico_lettres() et BM() dans un fichier que vous nommerez rechercheTextuelleBM.py.
- Rajouter un compteur de comparaisons effectuées.
- Ecrire également dans ce fichier, le code de la fonction recherche_naive() avec 2 boucles while et comparaison en commençant par la fin du motif. Y rajouter aussi un compteur de comparaisons.
- Traiter les cas suivants, à partir du texte correspondant au livre « Le rouge et le noir »
Et remplir le tableau ci-dessous (résultats à mettre en commentaire dans le fichier .py) :

	Script avec 2 boucles while en partant de la fin du motif	Algorithme de Boyer-Moore
Nombre de comparaisons pour le motif : motif = "a"		
Nombre de comparaisons pour le motif : motif = "Julien"		
Nombre de comparaisons pour le motif : motif = "peut facilement observer le moment de leur rentrée à l'hôtel"		
Nombre de comparaisons pour le motif ci-dessous, à copier-coller		

motif = ""
Il faut savoir que le curé de Verrières vieillard de quatre-vingts ans, mais qui devait à l'air vif de ces montagnes une santé et un caractère de fer, avait le droit de visiter à toute heure la prison, l'hôpital et même le dépôt de mendicité. C'était précisément à six heures du matin que M. Appert qui de Paris était recommandé au curé, avait eu la sagesse d'arriver dans une petite ville curieuse. Aussitôt il était allé au presbytère.""

- Conclure en répondant aux questions suivantes (en commentaire dans le fichier .py) de manière précise, la réponse attendue doit être argumentée et rédigée :
 - Question 1 : Dans quel cas de figure, l'algorithme de Boyer-Moore est-il plus performant
 - Question 2 : Quelle est finalement la complexité de cet algorithme ?