

I- Approche naïve-> <https://youtu.be/EeHwu5kl6II> (de 0 à 3mn)

```

1  #
2  # Recherche d'une chaîne de caractère
3  # Approche naïve
4  # Compléter le programme
5
6  texte="Je vais acheter de beaux poissons chez le poissonnier"
7  motif="poisson"
8
9  longueur_texte = len(texte)
10 longueur_motif = len(motif)
11
12 i = # Initialisation
13
14
15 while (i <= .....):
16
17     j= #initialisation
18
19     while j< ..... and (texte[ ] == motif[ ]):
20
21         j = ..... # Incrémenter j
22
23
24     if j == ..... :
25
26         print ("Motif trouvé position :",i)
27
28
29     i = i + .....
30

```

Insérer ligne 16 les « print » ci-dessous afin d'afficher le texte le motif avec les décalages successifs

```

print()
print (" ",texte)
print (" "*i,motif,end=" ")
print()

```

A partir des informations de la vidéo, compléter le programme de recherche « naïf ».

Combien de tests sont effectués dans le pire des cas ?

En déduire la complexité.

motif[4] motif[5] motif[6]
 motif = p o i s s o n
 motif[0] motif[1] motif[2] motif[3]

Je vais acheter de beaux poissons chez le poissonnier
 poisson

Je vais acheter de beaux poissons chez le poissonnier
 poisson

Je vais acheter de beaux poissons chez le poissonnier
 poisson

Motif trouvé position : 25

Je vais acheter de beaux poissons chez le poissonnier
 poisson

Je vais acheter de beaux poissons chez le poissonnier
 poisson

II- Algorithme de Boyer-Moore – Horspool

-> Voir vidéo Boyer-Moore (de 3mn à 7mn30)

```

1  #
2  # Recherche d'une chaîne de caractère
3  # Algo boyer moore (adapté en python :)
4  # B. HOAREAU
5
6  nb_caracteres = 256 # table des codes ascii
7  texte="Je vais acheter de beaux poissons chez le poissonnier"
8  motif="poisson"
9
10 longueur_texte = len(texte)
11 longueur_motif = len(motif)
12
13 table_saut = [longueur_motif]*256
14
15 for i in range(longueur_motif-1):
16     table_saut[ord(motif[i])] = longueur_motif - i - 1
17
18
19 i = 0
20 while (i <= longueur_texte - longueur_motif):
21     j= longueur_motif-1
22
23     print()
24     print (" ",texte)
25     print (" "*i,motif,end=" ")
26     print()
27
28     while j>=0 and (texte[i + j] == motif[j]):
29         j = j-1
30
31     if j<0:
32         print ("Motif trouvé position :",i)
33
34     i = i+ table_saut[ord(texte[i+longueur_motif-1])]
35

```

Pour créer une table de saut, on utilise le code ASCII des caractères comme index.

Par exemple ord(« p ») = 112
 Et table_saut[112] = 6

Par défaut, la table de saut est initialisée avec la valeur max du décalage = longueur_motif (7 ici)

> **Taper le programme** « Boyer Moore » et tester le.

Comparez avec l'algo naïf, que pouvez en vous conclure ?

Compréhension du programme.

Afin de comprendre un peu mieux ce que fait le programme, vous **testerez** les instructions séparément si nécessaire ou bien vous insérez judicieusement des « print » dans le programme.

* Que fait l'instruction « len » dans len(texte) par exemple ?

* Que fait l'instruction python ci-dessous ?

```
table_saut = [longueur_motif]*256
```

* A partir du code programme ci-dessous, compléter le tableau. Bien faire attention pour la valeur de i en fin de boucle, sachant que longueur_motif = 7

```
for i in range(longueur_motif-1):
    table_saut[ord(motif[i])] = longueur_motif - i - 1
```

i	Lettre	Code ASCII	Décalage
0	p	112	6
1	o		
2	i		
3	s		
4	s		
5	o		
6	n		7
	autre	X	7

Le 'o' et le 's' apparaissent plusieurs fois, quelles valeurs de décalages sont conservées dans ces cas là ?

* Si longueur_texte = 100 et longueur_motif = 10, pour quelle valeur de i on n'entrera plus dans la boucle ci-dessous : ?

```
19 i = 0
20 while (i <= longueur_texte - longueur_motif):
21     j = longueur_motif-1
```

* Que fait le code ci-dessous : ?

```
28 while j >= 0 and (texte[i + j] == motif[j]):
29     j = j - 1
30
31 if j < 0:
32     print ("Motif trouvé position :", i)
```

* L'instruction suivante permet de calculer le décalage à faire :

```
34 i = i + table_saut[ord(texte[i+longueur_motif-1])]
```

Je vais acheter de beaux poissons chez le poissonnier
poisson

Que vaut le prochain décalage dans le cas de figure ci-contre ?