

Informatique TB

Corrigé

2022

I Recherche de sous-chaîne par force brute

1. :

"AAT" apparait aux indices 2 et 7

2. :

La fonction correcte est chaines_egales2.

Explications non demandées :

fonction 1 compare des caractères des deux chaînes qui ne sont pas à la même position

fonction 3 dès que les deux chaînes partagent le même premier caractère le résultat renvoyé est True, par exemple chaines_egales3 ("AAT" , "ATT") renvoie True

fonction 4 même type de problème que pour la fonction 1.

- ```
3. def liste_occurences(texte,mot):
 L=[] # si aucune occurrence n'est trouvé on renvoie la liste vide
 n=len(mot)
 for i in range(len(texte)-n+1): #attention
 if chaines_egales2(texte[i:i+n], mot):
 L.append(i)
 return L

4. def premiere_occurence(texte,mot):
 L=liste_occurences(texte,mot)
 if L==[]:
 return None
 else:
 return min(L) #on peut aussi renvoyer L[0]
```

## II Bord d'une chaîne de caractères

1.

Le bord de la chaîne de caractères "ATTACGTTCCATTAC" est "ATTAC".

- ```
2. def longueur_bord (ch ):
    n = len(ch)
    jmax = 0
    for j in range (1,n): #n car la chaîne entière ne peut pas être le bord
        if chaines_egales(ch [:j],ch [n-j:]):
            jmax = j
    return jmax
```

3.

```
def longueurs_bords_prefixes(ch):  
    R=[] #résultat  
    for i in range(len(ch)):  
        R.append(longueur_bord(ch[:i+1]))  
    return R
```

4. "AATGCAAT" est un préfixe de la chaîne donné en exemple on peut reprendre les premières étapes du raisonnement donné en exemple, en s'arrêtant à la ligne "AAT" est le bord de la chaîne "AATGCAAT"

La commande `longueurs_bords_prefixes("AATGCAAT")` renvoie `[0, 1, 0, 0, 0, 1, 2, 3]`.

III Recherche de sous-chaîne par l'algorithme de Knuth-Morris-Pratt(KMP)

```
def kmp(texte , mot ):  
    n = len( texte )  
    p = len(mot)  
    B = longueurs_bords_prefixes (mot)  
    i = 0  
    j = 0  
    while i <= len(texte)-len(mot) :  
        while j < len(mot) and mot [j] == texte [i+j]:  
            j += 1  
        if j == len(mot) :  
            return i  
        if j == 0 :  
            i = i+1  
        else:  
            i = i+j-B[j-1]  
            j = B[j-1]  
    return None
```