
RECHERCHE TEXTUELLE

Exercice 0 À partir de la définition de la fonction `recherche(m, t)` implémentant un algorithme naïf de recherche du motif `m` dans le texte `t`, détermine le nombre de comparaisons de caractères effectuées lors de l'appel de :

```
recherche('chercher', 'chercher, rechercher et chercher encore')
```

```
1 def recherche(m, t):
2     for i in range(len(t)-len(m)+1):
3         j = 0
4         while j < len(m) and m[j] == t[i+j] :
5             j += 1
6         if j == len(m):
7             print("occurrence trouvé à la position", i)
```

Exercice 1 Construire à la main a table de décalages pour le motif `'chercher'`.

Exercice 2 En utilisant la table obtenue à l'exercice précédent, dérouler manuellement l'exécution de la fonction `recherche_bm(m, t)` lors de l'appel de

```
recherche_bm('chercher', 'chercher, rechercher et chercher encore')
```

en donnant notamment :

- les valeurs successives de l'indice `i` ;
- le nombre total de comparaisons de caractères.

Compare le nombre de comaparaisons obtenu avec celui de l'exercice 0.

```
1 def recherche(m, t):
2     d = table_bm(m)
3     i = 0
4     while i <= len(t) - len(m):
5         j = len(m)-1
6         different = False
7         while j >= 0 and not different:
8             if m[j] == t[i+j]:
9                 j -= 1
10            else:
11                different = True
12            if different:
13                i += decalage(d, j, t[i+j])
14            else:
15                print("occurrence trouvé à la position", i)
16                i += 1
```