

Corrigé — Mo-Mo-Motifs

I Algorithmes de cours

QUESTION 1. *Comptage*

coeff
1

QUESTION 2. *Création d'une matrice $n \times m$*

coeff
1

II Mot de 7 lettres

QUESTION 3. *Hachage caractère*

coeff recherche décalage
1 6 3

```
def hord(c) :  
    for i in range(26) :  
        if minascii[i] == c :  
            return i + 1  
    return -1
```

QUESTION 4. *Hachage par somme*

coeff comptage appel
1 6 3

```
def hsomme(s) :  
    h = 0  
    for c in s :  
        h += hord(c)  
    return h
```

QUESTION 5. *Hachage égaux*

coeff
1

nadine et dianne

QUESTION 6. Hachage séquentiel

coeff
1

lacier s'obtient depuis place en enlevant le p du tête et en rajoutant r en queue de chaîne. Le haché par somme de lacier s'obtient donc en soustrayant hord('p') et en ajoutant hord('r').

III Mot de 7 lettres

On rappelle le principe de l'algorithme de la fenêtre glissante.

Pour chercher une chaîne s de longueur n dans un texte txt, on délimite une fenêtre de recherche dans txt de longueur n, initialement débutant à l'indice 0 et qui se déplace d'indice en indice tant que la fenêtre ne fait pas apparaître exactement s.

Par exemple pour chercher lit dans paleolithique, on crée une fenêtre qui compare successivement leo à pal, ale, leo, eol, oli, lit.

QUESTION 7. Fenêtre différente

coeff recherche résultat
2 6 3

```
def difference(s,txt,i) :
    # Parcours inverse de s
    for k in range(len(s)-1, -1, -1) :
        if txt[i+k] != s[k] :
            return k
    return -1
```

QUESTION 8. Fenêtre glissante

coeff recherche borne appel
2 3 3 3

```
def rgissante(s,txt) :
    for i in range(len(txt)-len(s)+1) :
        if difference(s,txt,i) == -1 :
            return i
    return -1
```

QUESTION 9. Fenêtre hachée

coeff Trou 1 Trou 2 Trou 3
2 2 2 5

```
def rabin_karp(s,txt) :
    n, m = len(s), len(txt)
    hs, hf = hsomme(s), hsomme(txt[0:n])
    for i in range(m-n+1) :
        if hs == hf :
            if difference(s,txt,i) == -1 :
                return i
        if i+n < m :
            hf = hf - hord(txt[i]) + hord(txt[i+n])
    return -1
```

IV Mot en 5 lettres

QUESTION 10. Saut dans un motif

coeff	explication	1	5	6
2	4	2	1	2

La fenêtre 2 correspond à leo. Le o en dernière position n'est pas dans l it donc les fenêtre 3 et 4 qui contiennent toujours ce o ne peuvent pas correspondre. On peut sauter directement à la fenêtre 5 qui ne contient pas le o.

- Pour $i = 1$: le l final de aleol apparaît à la position 2 de s et plus après. Donc la prochaine fenêtre utile est celle qui place ce l en face de celui de la position 2 de s, donc la fenêtre 3.
- Pour $i = 5$: avec le même raisonnement on se place en fenêtre 6.
- Pour $i = 6$: le q n'est pas dans s et il ne reste que 2 caractères après le q, il n'y plus de fenêtre utile à vérifier.

QUESTION 11. Table de saut du motif - Exemple

coeff	valeurs lettre	cas du e
1	7	2

Table de saut de "denadine".

0	1...2	3	4	5...7	8	9...12	13	14 ... 25
4	8	3	6	8	2	8	1	8

QUESTION 12. Table de saut du motif

coeff	dernière occurrence	remplissage
1	3	6

```
def ts_motif(s) :
    n = len(s)
```

```
ts = [ n ] * 26
for i in range(n-1) :
    h = hord(s[i])
    ts[h-1] = n-i-1
return ts
```

QUESTION 13. *Recherche de motif*

coeff	parcours	saut	resultat
2	2	3	4

```
def horspool(s,txt) :
    n, m = len(s), len(txt)
    ts = ts_motif(s)
    i = 0
    while i < m - n + 1 :
        if difference(s,txt,i) == -1 :
            return i
        i += ts[hord(txt[i+n-1])-1]
    return -1
```