

Informatique PCSI

Corrigé TP 2

Exercices

Exercice 1

La variable i varie de 0 à 4. Si i vaut 0, j varie de 1 à 4, donc cela donne 4 appels. Puis nous avons 3 appels, puis 2 et enfin 1 appel. Le total est $4 + 3 + 2 + 1 = 10$ appels.

Exercice 2

1. La variable i prend la valeur 0, et il y a une addition.

La variable j prend successivement les valeurs 0, 1, 2, avec une addition à chaque passage, donc cela fait 3 additions.

La variable i prend la valeur 1, il y a une addition.

La variable j prend successivement les valeurs 0, 1, 2, donc cela fait encore 3 additions.

Le total est $2 + 2 \times 3 = 8$ additions.

2. On fait le même type de calcul que pour la question 1 en remplaçant le 3 de la boucle interne par n , donc le total est $2 + 2n = 2(n+1)$ additions.
3. Cette fois on remplace le 2 par n donc le total est : $n + n \times n = n(n+1)$ additions.

Exercice 3

Les valeurs de x sont strictement décroissantes et la valeur de y est remise à 0 dès qu'elle n'est plus strictement inférieure à celle de x . Au dernier passage dans la boucle interne, y vaut 0, et x vaut 1 : y prend alors la valeur 1 et x la valeur 0; on sort de la boucle interne puis de la boucle externe.

On a une boucle infinie si on remplace $y = 0$ par $y = 1$.

Exercice 4

Les comparaisons sont marquées par les caractères en gras du motif et leur nombre figure dans la colonne de gauche. Dès qu'une comparaison échoue, le motif est décalé à chaque fois d'une unité vers la droite.

	a	b	a	b	a	b	a	b	a	b
3	a	b	c							
1		a	b	c						
3			a	b	c					
1				a	b	c				
3					a	b	c			
1						a	b	c		
3							a	b	c	
1								a	b	c
2									a	b

Le total est de 18 comparaisons, $16 + 2$. Les 16 sont obtenues par 4 fois $(3 + 1)$ ou 8 fois 2 (en moyenne nous avons 2 comparaisons par caractère du texte) puis les 2 dernières comparaisons avec le motif qui déborde du texte. Le total est donc de 9×2 comparaisons.

En suivant le même principe, pour un texte de 100 caractères, nous avons 98 fois 2 comparaisons puis les 2 dernières avec le motif qui déborde du texte, soit 99×2 comparaisons.

Avec n caractères, nous avons $(n - 1) \times 2$ comparaisons, donc de l'ordre de $2n$ comparaisons.

Exercice 5

On écrit le code de la fonction recherche du document préparatoire, puis le code ci-dessous :

```
from random import randint
choix = "ACGT"

texte = [choix[randint(0, 3)] for i in range(10000)]
motif = "CAGCAG"
print(recherche(texte, motif))
```

Sur quelques essais, on trouve en moyenne trois fois le motif.

Exercice 6

On définit une distance entre deux points, (ici la distance entre deux réels), et une liste de points (ici des nombres réels) :

```
def d(x, y):
    return abs(x - y)

E = list(range(1000))
```

```
def proches_voisins(E, x, k, d):
    voisins = []
    for i in range(k):
        voisins.append(E[i])
    for i in range(k, len(E)):
        dist = d(x, E[i])
        u = i
        for j in range(k): # recherche d'un maximum
            if d(voisins[j], x) > dist:
                dist = d(voisins[j], x)
                u = j
        if u != i:
            voisins[u] = E[i]
    return voisins
```

On peut tester le programme dans l'interpréteur :

```
>>> proches_voisins(E, 15.2, 1, d)
[15]
>>> proches_voisins(E, 15.2, 4, d)
[16, 17, 14, 15]
```

Un problème de lenteur apparaît clairement.