

Algorithmes de tri – Activités préparatoires.

Les activités préparatoires sont disponibles aux liens suivants : sujet – <https://bit.ly/3eRFYEB> et corrige – <https://bit.ly/3eIwFa8>.

QCM

Question 1 Combien d'échanges effectue la fonction Python suivante pour trier un tableau de 10 éléments au pire des cas ?

```
def tri(tab) :
    for i in range (1, len(tab)) :
        for j in range (len(tab) - i) :
            if tab[j] > tab[j+1] :
                tab[j], tab[j+1] = tab[j+1], tab[j]
```

- 1°) 45.
- 2°) 100.
- 3°) 10.
- 4°) 55.

Question 2 Que vaut l'expression $f([7, 3, 1, 8, 19, 9, 3, 5], 0)$?

```
def f(t,i) :
    im = i
    m = t[i]
    for k in range(i+1, len(t)) :
        if t[k] < m :
            im, m = k, t[k]
    return im
```

- 1°) 1.
- 2°) 2.
- 3°) 3.
- 4°) 4.

Question 3 Laquelle de ces listes de chaînes de caractères est triée en ordre croissant ?

- 1°) ['Chat' , 'Cheval' , 'Chien' , 'Cochon'].
- 2°) ['Cochon' , 'Chat' , 'Cheval' , 'Chien'].
- 3°) ['Cheval' , 'Chien' , 'Chat' , 'Cochon'].
- 4°) ['Chat' , 'Cochon' , 'Cheval' , 'Chien'].

Question 4 Laquelle de ces listes de chaînes de caractères est triée en ordre croissant ?

- 1°) ['12' , '142' , '21' , '8'].
- 2°) ['8' , '12' , '142' , '21'].
- 3°) ['8' , '12' , '21' , '142'].
- 4°) ['12' , '21' , '8' , '142'].

Question 5 Quelle est la valeur de la variable `table` après exécution du programme Python suivant ?

```
table = [12, 43, 6, 22, 37]
for i in range(len(table) - 1):
    if table[i] > table[i+1] :
        table[i] , table[i+1] = table[i+1] , table[i]
```

- 1°) [12, 6, 22, 37, 43].
- 2°) [6, 12, 22, 37, 43].
- 3°) [43, 12, 22, 37, 6].
- 4°) [43, 37, 22, 12, 6].

Question 6 Un algorithme cherche la valeur maximale d'une liste non triée de taille n . Combien de temps mettra cet algorithme sur une liste de taille $2n$?

- 1°) Le même temps que sur la liste de taille n si le maximum est dans la première moitié de la liste.
- 2°) On a ajouté n valeurs, l'algorithme mettra donc n fois plus de temps que sur la liste de taille n .
- 3°) Le temps sera simplement doublé par rapport au temps mis sur la liste de taille n .
- 4°) On ne peut pas savoir, tout dépend de l'endroit où est le maximum.

Question 7 On souhaite écrire une fonction `tri_selection(t)`, qui trie le tableau `t` dans l'ordre croissant : parmi les 4 programmes suivants, lequel est correct ?

```
def tri_selection(t) :
    for i in range (len(t)-1) :
        min = i
        for j in range(i+1,len(t)):
            if t[j] < t[min]:
                min = j
        tmp = t[i]
        t[i] = t[min]
        t[min] = tmp
def tri_selection(t) :
    for i in range (len(t)-1) :
        min = i
        for j in range(i+1,len(t)-1):
            if t[j] < t[min]:
                min = j
        tmp = t[i]
        t[i] = t[min]
        t[min] = tmp
def tri_selection(t) :
    for i in range (len(t)-1) :
        min = i
        for j in range(i+1,len(t)):
            if t[j] < min:
                min = j
        tmp = t[i]
        t[i] = t[min]
        t[min] = tmp
def tri_selection(t) :
    for i in range (len(t)-1) :
        min = i
        for j in range(i+1,len(t)):
            if t[j] < t[min]:
                min = j
        tmp = t[i]
        t[min] = t[i]
        t[i] = tmp
```

- 1°) Fonction 1.
- 2°) Fonction 2.
- 3°) Fonction 3.
- 4°) Fonction 4.

Question 8 De quel type de tri s'agit-il ?

```
def tri(lst):
    for i in range(1,len(lst)):
        valeur = lst[i]
        j = i
        while j>0 and lst[j-1]>valeur:
            lst[j]=lst[j-1]
            j = j-1
        lst[j]=valeur
```

- 1°) Tri par insertion.

- 2°) Tri fusion.
- 3°) Tri par sélection.
- 4°) Tri à bulles.

Question 9 De quel type de tri s'agit-il ?

```
def tri(lst):
    nb = len(lst)
    for i in range(0,nb):
        ind_plus_petit = i
        for j in range(i+1,nb) :
            if lst[j] < lst[ind_plus_petit] :
                ind_plus_petit = j
        if ind_plus_petit is not i :
            temp = lst[i]
            lst[i] = lst[ind_plus_petit]
            lst[ind_plus_petit] = temp
```

- 1°) Tri par insertion.
- 2°) Tri fusion.
- 3°) Tri par sélection.
- 4°) Tri à bulles.

Question 10 Un algorithme est en complexité quadratique. Codé en python, son exécution pour des données de taille 100 prend 12 millisecondes. Si l'on fournit des données de taille 200 au programme, on peut s'attendre à un temps d'exécution d'environ :

- 1°) 48 millisecondes.
- 2°) 24 millisecondes.
- 3°) 12 millisecondes.
- 4°) 96 millisecondes.

Question 11 À quel type de tri correspond l'invariant de boucle ci-dessous :

- tous les éléments d'indices 0 à $i-1$ sont déjà triés,
- tous les éléments d'indices i à n sont de valeurs supérieures à ceux de la partie triée.

- 1°) Tri par insertion.
- 2°) Tri fusion.
- 3°) Tri par sélection.
- 4°) Tri à bulles.

Question 12 Quel est l'invariant de boucle qui correspond précisément à cet algorithme ?

On considère un algorithme de tri par sélection, dans lequel la fonction `echanger(tab[i], tab[j])` effectue l'échange des i ème et j ème valeurs du tableau `tab`.

```

nom: tri_sélection

paramètre: tab, tableau de n entiers, n>=2

Traitement:
pour i allant de 1 à n-1:
    pour j allant de i+1 à n:
        si tab[j] < tab[i]:
            échanger(tab[i], tab[j])
renvoyer tab

```

- 1°) Tous les éléments d'indice supérieur ou égal à i sont triés par ordre croissante.
- 2°) Tous les éléments d'indice compris entre 0 et i sont triés et les éléments d'indice supérieurs ou égal à i leurs sont tous supérieurs.
- 3°) Tous les éléments d'indice supérieur ou égal à i sont non triés.
- 4°) Tous les éléments d'indice compris entre 0 et i sont triés, on ne peut rien dire sur les éléments d'indice supérieur ou égal à i.

Question 13 *Quel est le type de tri qui correspond à cet algorithme ?*

```

nom: tri_mystere

paramètre: tab, tableau de n entiers, non trié,
           non vide

Traitement:
pour i allant de 1 à n-1:
    pour j allant de i+1 à n:
        si tab[j] < tab[i]:
            échanger(tab[i], tab[j])
renvoyer tab

```

- 1°) Tri par insertion.
- 2°) Tri fusion.

- 3°) Tri par sélection.
- 4°) Tri rapide.

Question 14 *Quel est l'invariant de boucle qui correspond précisément à cet algorithme ?*

```

nom: tri_insertion

paramètre: tab, tableau de n entiers, n >= 2

Traitement:
pour i allant de 2 à n:
    j = i
    tant que j > 1 et tab[j-1] > tab[j]:
        échanger(tab[j-1], tab[j])
        j = j-1
renvoyer tab

```

- 1°) Tous les éléments d'indice compris entre 0 et i sont triés et les éléments d'indice supérieurs ou égal à i leurs sont tous supérieurs.
- 2°) Tous les éléments d'indice supérieur ou égal à i sont triés par ordre croissant.
- 3°) Tous les éléments d'indice compris entre 0 et i sont triés, on ne peut rien dire sur les éléments d'indice supérieur ou égal à i.
- 4°) Tous les éléments d'indice supérieur ou égal à i sont non triés par ordre croissants.

Question 15 *Parmi les propositions suivantes, quelle est celle qui ne correspond pas à une méthode de tri ?*

- 1°) Par sélection.
- 2°) Par insertion.
- 3°) Par rotation.
- 4°) Par fusion.