

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
30/05/2023	8 – Tris	INT – Sujet

Note

## Tris

Nom :

Prénom :

### Exercice 1: Tri insertion en place

Question 1: En quelques mots, décrire la méthode du tri proposé

1-1

Question 2: Compléter l'algorithme suivant afin qu'il réalise ce tri stable

```
def tri_insertion(L):
    for i in range(len(L)):
        while (i-1 >= 0)
            L[i], L[i-1] = L[i-1], L[i]
```

1-2

Question 3: Préciser la condition à respecter pour que ce tri soit stable ?

Ecrire ci-dessous la condition du code ci-dessus pour que le tri soit	
Stable	Non stable

1-3

Question 4: Compléter le tableau ci-dessous à propos de l'algorithme proposé

	Meilleur des cas	Pire des cas
Spécificités de L		
Complexité en temps pour n termes		

1-4

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
30/05/2023	8 – Tris	INT – Sujet

## Exercice 2: Tri sélection en place

Question 1: En quelques mots, décrire la méthode du tri proposé

2-1

Question 2: Compléter l'algorithme suivant afin qu'il réalise ce tri

```
def tri_selection(L):
    N = len(L)
    for i in range(N-1):
        im = i

        for j in range(im+1, N):
            if L[j] < L[im]:
                im = j
```

2-2

Question 3: Compléter le tableau ci-dessous à propos de la stabilité de ce tri

Ecrire ci-dessous la condition du code ci-dessus pour que le tri soit	
Stable	Non stable
Obtenir la stabilité revient à sélectionner le ..... minimum	

2-3

Question 4: Préciser, détaillant précisément le calcul, la complexité en temps de cet algorithme pour une liste de n termes

Calcul réalisé	Complexité

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
30/05/2023	8 – Tris	INT – Sujet

### Exercice 3: Tri rapide avec listes auxiliaires

Question 1: En quelques mots, décrire la méthode du tri proposé

3-1

Question 2: Compléter l'algorithme suivant afin qu'il réalise ce tri stable

```
def tri_rapide(L):
    if
    else:
        L1,LP,L2 = [],[L[0]],[]
        for i in range(1,len(L)):
            if
                L1.append(L[i])
            else:
                L2.append(L[i])

        return
```

3-2

Question 3: Préciser la condition à respecter pour que ce tri soit stable ?

Ecrire ci-dessous la condition du code ci-dessus pour que le tri soit	
Stable	Non stable
Obtenir la stabilité revient à mettre les exæquo au pivot dans .....	

3-3

Question 4: Compléter le tableau ci-dessous à propos de l'algorithme proposé

	Meilleur des cas	Pire des cas
Spécificités du pivot à chaque étape		
Complexité en temps pour n termes		

3-4

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
30/05/2023	8 – Tris	INT – Sujet

## Exercice 4: Tri fusion avec listes auxiliaires

Question 1: En quelques mots, décrire la méthode du tri proposé

4-1

Question 2: Compléter l'algorithme suivant afin qu'il réalise la fusion ordonnée « stable »

```
def fusion_ordonnee(L1,L2):
    Fusion = []
    i1 = i2 = 0
    while i1 < len(L1) and i2 < len(L2):
        if
            Fusion.append(L1[i1])
            i1 += 1
        else:
            Fusion.append(L2[i2])
            i2 += 1
    Fusion +=
    return Fusion
```

4-2

Question 3: Compléter l'algorithme suivant afin qu'il réalise ce tri

```
def tri_fusion(L):
    if len(L) <= 1:
        return L
    else:
        Im = len(L)//2
        L1 = L[0:Im]
        L2 = L[Im:len(L)]

        return
```

4-3

Question 4: Préciser la condition à respecter pour que ce tri soit stable ?

Ecrire ci-dessous la condition du code ci-dessus pour que le tri soit	
Stable	Non stable
Obtenir la stabilité revient à mettre les exæquo dans .....	

4-4

Préciser la complexité en temps de cet algorithme pour une liste de n termes

4-5

Dernière mise à jour	Informatique	Denis DEFAUCHY – <a href="#">Site web</a>
30/05/2023	8 – Tris	INT – Sujet

## Exercice 5: Tri par comptage

On se limite à une liste de  $n$  entiers positifs.

**Question 1:** En quelques mots, décrire la méthode du tri proposé

5-1

**Question 2:** Compléter l'algorithme suivant afin qu'il réalise ce tri

```
def tri_comptage(L):
    Res = []
    if len(L) >= 1:
        m = L[0]
        for t in L:
            if t > m:

        LP =
        for i in range(len(L)):
            LP[L[i]] += 1
        for i in range(len(LP)):
            Res +=
    return Res
```

5-2

Remarque : `>>> 0*[3]`  
`[]`

**Question 3:** Préciser, en justifiant brièvement la réponse, la complexité en temps de cet algorithme pour une liste de  $n$  termes de maximum  $m$

5-3