

Tris.

2013

Nature de ce document

- Ce pdf est un diaporama. Si vous n'avez pas encore installé okular sur votre clef, il est temps de le faire !
- Visualisez ce document avec okular en basculant en mode présentation (view/présentation).

- Il existe de nombreux algorithmes de tris.
- Leur utilisation dans la pratique dépend :
 - de la structure des données à trier,
 - de l'efficacité de l'algorithme utilisé : nous verrons en effet que pour une même structure, certains algorithmes sont beaucoup plus lents que d'autres.

Connaissances à acquérir.

A l'issue du travail sur ce document, vous **devez connaître les deux algorithmes de tris suivants et savoir les programmer** :

- tri par sélection,
- tri par fusion.

Travail à réaliser

Le document contient 3 exercices à déposer dans les casiers numériques de vos enseignants avant le 18/03/2013.

Intermède sur la portée des variables



Python

```
def f(a,b) :  
    a,b=b,a  
    print a,b  
  
a,b=3,5  
f(a,b)  
print a,b
```

Quels sont les affichages à l'exécution de ce code ?

Les affichages

Affichages :

- 5 3 variables a et b internes à la fonction.
- 3 5 variables a et b globales.

Intermède sur la portée des variables et l'affectation



Python

```
def h(L) :  
    L[0],L[1]=L[1],L[0]  
    print L  
  
L=[4,7,9]  
h(L)  
print L
```

Quels sont les affichages à l'exécution de ce code ?

Affichages

Les affichages :

[7, 4, 9]

[7, 4, 9]

Intermède sur la portée des variables et l'affectation



Python

```
def g(M) :  
    M[0],M[1]=M[1],M[0]  
    print M  
  
L=[4,7,9]  
g(L)  
print L
```

Quels sont les affichages à l'exécution de ce code ?

Affichages

Les affichages :

[7, 4, 9]

[7, 4, 9]

Tri par sélection

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

- ① Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- ② Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- ③ Reprendre au point 1 tant que des éléments sont noirs.

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

1	3	7	8	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

- ① Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- ② Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- ③ Reprendre au point 1 tant que des éléments sont noirs.

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

1	3	7	8	5	6	8	4	7	9
1	3	7	8	5	6	8	4	7	9

- ➊ Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- ➋ Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- ➌ Reprendre au point 1 tant que des éléments sont noirs.

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

1	3	7	8	5	6	8	4	7	9
1	3	7	8	5	6	8	4	7	9
1	3	4	8	5	6	8	7	7	9

- 1 Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- 2 Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- 2 Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

1	3	7	8	5	6	8	4	7	9
1	3	7	8	5	6	8	4	7	9
1	3	4	8	5	6	8	7	7	9
1	3	4	5	8	6	8	7	7	9

Présentation de l'algorithme de tri par sélection.

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément minimum.
- 2 Échanger le minimum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

1	3	7	8	5	6	8	4	7	9
1	3	7	8	5	6	8	4	7	9
1	3	4	8	5	6	8	7	7	9
1	3	4	5	8	6	8	7	7	9
1	3	4	5	6	8	8	7	7	9
1	3	4	5	6	7	8	8	7	9
1	3	4	5	6	7	7	8	8	9
1	3	4	5	6	7	7	8	8	9
1	3	4	5	6	7	7	8	8	9
1	3	4	5	6	7	7	8	8	9

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.
- 0, 2, 7, 15, 8, 3.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.
- 0, 2, 7, 15, 8, 3.
- 0, 2, 3, 15, 8, 7.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.
- 0, 2, 7, 15, 8, 3.
- 0, 2, 3, 15, 8, 7.
- 0, 2, 3, 7, 8, 15.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.
- 0, 2, 7, 15, 8, 3.
- 0, 2, 3, 15, 8, 7.
- 0, 2, 3, 7, 8, 15.
- 0, 2, 3, 7, 8, 15.

Exercice corrigé 1.

Sur papier, donner les étapes d'un tri par sélection pour le tri en ordre croissant de la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 0, 2, 7, 15, 8, 3.
- 0, 2, 7, 15, 8, 3.
- 0, 2, 3, 15, 8, 7.
- 0, 2, 3, 7, 8, 15.
- 0, 2, 3, 7, 8, 15.
- 0, 2, 3, 7, 8, 15.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :

Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.
- 15, 8, 7, 2, 0, 3.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 2, 0, 3.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 3, 0, 2.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 3, 0, 2.
- 15, 8, 7, 3, 2, 0.

Exercice corrigé 2.

On modifie le principe algorithmique de la façon suivante :
Initialisation : les éléments prennent tous la couleur noire.

- 1 Parcourir les éléments noirs, retenir le rang d'un élément maximum.
- 2 Échanger le maximum avec la tête noire et lui donner la couleur rouge.
- 3 Reprendre au point 1 tant que des éléments sont noirs.

Sur papier, donner les étapes de ce tri par sélection pour la liste : $L=[8, 2, 7, 15, 0, 3]$.

Correction.

- 15, 2, 7, 8, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 2, 0, 3.
- 15, 8, 7, 3, 0, 2.
- 15, 8, 7, 3, 2, 0.
- 15, 8, 7, 3, 2, 0.

Exercice à rendre numéro 1.

Traduire l'algorithme ci-dessous, en respectant les contraintes décrites, en langage python.

- Entrée : une liste de nombres.
- Sortie : la même liste triée dans l'ordre croissant.
- Traitement : tri des éléments de la liste suivant le principe de sélection du minimum présenté ci-dessus. Ce traitement devra respecter les contraintes données ci-après.

Contraintes à respecter.

On décomposera le programme en plusieurs fonctions. Les entêtes de ces fonctions sont définies ci-après et devront être respectées.

Les fonctions à programmer.

Python

```
def echange(L,i,j) :
```

```
    """L est une liste, i et j sont des entiers compris entre 0 et  
    len(L)-1.
```

```
    La procédure échange les contenus de L[i] et L[j]"""
```

```
    TRAITEMENT
```

```
def selection(L,k) :
```

```
    """L est une liste, k est un entier compris entre 0 et len(L)-1.
```

```
    La procédure échange L[k] avec un élément min parmi  
    L[k],L[k+1],...,L[len(L)-1]"""
```

```
    TRAITEMENT
```

Les fonctions à programmer.



Python

```
def triselection(L) :
```

```
    """ L est la liste à trier.
```

```
    La fonction trie L suivant le principe du tri par sélection  
    en utilisant les deux fonctions précédentes. """
```

```
    TRAITEMENT
```

Tri par fusion

Le principe de la fusion

J'ai devant moi deux paquets de cartes triées A et B, la plus petite carte sur le dessus.

Pour fusionner les deux paquets en un seul paquet trié :

Le principe de la fusion

J'ai devant moi deux paquets de cartes triées A et B, la plus petite carte sur le dessus.

Pour fusionner les deux paquets en un seul paquet trié :

- prendre la plus petite carte visible sur les paquets A, B et la déposer pour commencer un paquet C.

Le principe de la fusion

J'ai devant moi deux paquets de cartes triées A et B, la plus petite carte sur le dessus.

Pour fusionner les deux paquets en un seul paquet trié :

- prendre la plus petite carte visible sur les paquets A, B et la déposer pour commencer un paquet C.
- prendre la plus petite carte visible sur les paquets A, B et la glisser sous le paquet C.

Le principe de la fusion

J'ai devant moi deux paquets de cartes triées A et B, la plus petite carte sur le dessus.

Pour fusionner les deux paquets en un seul paquet trié :

- prendre la plus petite carte visible sur les paquets A, B et la déposer pour commencer un paquet C.
- prendre la plus petite carte visible sur les paquets A, B et la glisser sous le paquet C.
- recommencer le point précédent tant que les paquets A et B sont non vides.

Le principe de la fusion

J'ai devant moi deux paquets de cartes triées A et B, la plus petite carte sur le dessus.

Pour fusionner les deux paquets en un seul paquet trié :

- prendre la plus petite carte visible sur les paquets A, B et la déposer pour commencer un paquet C.
- prendre la plus petite carte visible sur les paquets A, B et la glisser sous le paquet C.
- recommencer le point précédent tant que les paquets A et B sont non vides.
- glisser le reste du paquet A ou B non vide sous le paquet C.

Vocabulaire.

Convenons d'appeler segment d'une liste toute sous-liste d'éléments consécutifs.

Exemple.

Pour la liste $L = [w, j, e, k, p, b, u]$:

- $[j, e, k, p]$ est un segment de la liste.
- $[j, p]$ n'est pas un segment de la liste puisqu'on a omis deux éléments qui se trouvaient entre a et p dans la liste.

Le principe du tri par fusion

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Le principe du tri par fusion

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

- ① Fusionner les segments de longueur 1.
- ② Fusionner les segments de longueur 2.
- ③ Fusionner les segments de longueur 2^2 .
- ④ Fusionner les segments de longueur 2^3 .
- ⑤ ...
- ⑥ Fusionner les segments de longueur 2^k tant que $2^k < \text{longueur de la liste}$.

Le principe du tri par fusion

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

- ① Fusionner les segments de longueur 1.
- ② Fusionner les segments de longueur 2.
- ③ Fusionner les segments de longueur 2^2 .
- ④ Fusionner les segments de longueur 2^3 .
- ⑤ ...
- ⑥ Fusionner les segments de longueur 2^k tant que $2^k < \text{longueur de la liste}$.



Le principe du tri par fusion

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

- ① Fusionner les segments de longueur 1.
- ② Fusionner les segments de longueur 2.
- ③ Fusionner les segments de longueur 2^2 .
- ④ Fusionner les segments de longueur 2^3 .
- ⑤ ...
- ⑥ Fusionner les segments de longueur 2^k tant que $2^k < \text{longueur de la liste}$.

3	8	1	7	5	6	4	8	7	9
1	3	7	8	4	5	6	8	7	9

Le principe du tri par fusion

Objectif : trier les éléments de la liste :

8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

- ① Fusionner les segments de longueur 1.
- ② Fusionner les segments de longueur 2.
- ③ Fusionner les segments de longueur 2^2 .
- ④ Fusionner les segments de longueur 2^3 .
- ⑤ ...
- ⑥ Fusionner les segments de longueur 2^k tant que $2^k < \text{longueur de la liste}$.

3	8	1	7	5	6	4	8	7	9
1	3	7	8	4	5	6	8	7	9
1	3	4	5	6	7	8	8	7	9

Le principe du tri par fusion

Objectif : trier les éléments de la liste :

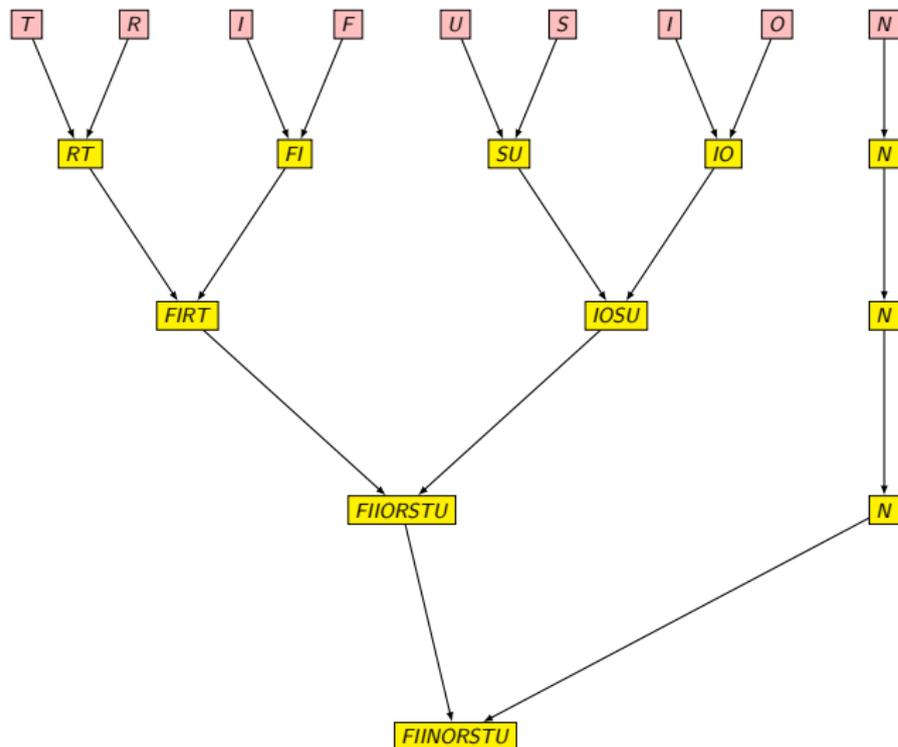
8	3	7	1	5	6	8	4	7	9
---	---	---	---	---	---	---	---	---	---

Principe algorithmique.

- ① Fusionner les segments de longueur 1.
- ② Fusionner les segments de longueur 2.
- ③ Fusionner les segments de longueur 2^2 .
- ④ Fusionner les segments de longueur 2^3 .
- ⑤ ...
- ⑥ Fusionner les segments de longueur 2^k tant que $2^k < \text{longueur de la liste}$.

3	8	1	7	5	6	4	8	7	9
1	3	7	8	4	5	6	8	7	9
1	3	4	5	6	7	8	8	7	9
1	3	4	5	6	7	7	8	8	9

Illustration à l'aide d'un arbre



Exercice corrigé.

Donner les étapes d'un tri par fusion de la liste $L=[2, 8, 15, 1, 7, 4, 9, 21, 10]$.

Exercice corrigé.

Donner les étapes d'un tri par fusion de la liste $L=[2, 8, 15, 1, 7, 4, 9, 21, 10]$.

Résolution.

- 2, 8 , 1, 15, 4, 7, 9, 21 , 10

Exercice corrigé.

Donner les étapes d'un tri par fusion de la liste $L=[2, 8, 15, 1, 7, 4, 9, 21, 10]$.

Résolution.

- 2, 8 , 1, 15, 4, 7, 9, 21 , 10
- 1, 2, 8, 15 , 4, 7, 9, 21, 10

Exercice corrigé.

Donner les étapes d'un tri par fusion de la liste $L=[2, 8, 15, 1, 7, 4, 9, 21, 10]$.

Résolution.

- 2, 8 , 1, 15, 4, 7, 9, 21 , 10
- 1, 2, 8, 15 , 4, 7, 9, 21, 10
- 1, 2, 4, 7, 8, 9, 15, 21 , 10

Exercice corrigé.

Donner les étapes d'un tri par fusion de la liste $L=[2, 8, 15, 1, 7, 4, 9, 21, 10]$.

Résolution.

- 2, 8 , 1, 15, 4, 7, 9, 21 , 10
- 1, 2, 8, 15 , 4, 7, 9, 21, 10
- 1, 2, 4, 7, 8, 9, 15, 21 , 10
- 1, 2, 4, 7, 8, 9, 10, 15, 21

Programmation.

Pour programmer cette méthode de tri, on va décomposer les tâches en plusieurs fonctions.

- Une procédure "fusion" pour la fusion de deux segments triés de la liste.
- Une procédure "doublesegment" qui transformera une liste dont les segments de longueur k sont triés en une liste dont les segments de longueur $2k$ sont triés.
- La fonction de tri par fusion qui utilisera les deux procédures précédentes.

Programmation de la fusion.

Nous allons écrire une procédure fusion qui fusionne deux segments triés consécutifs de la liste L en un seul segment trié de la liste L .

Plus précisément, nous allons décrire une procédure $\text{fusion}(p,q,r,L)$ telle que :

- En entrée : p, q, r des entiers correspondants à des indices de la liste L . Les segments $[L_{p+1}, L_{p+2}, \dots, L_q]$ et $[L_{q+1}, L_{q+2}, \dots, L_r]$ de la liste L sont déjà triés dans l'entrée.
- En sortie : les segments $[L_{p+1}, L_{p+2}, \dots, L_q]$ et $[L_{q+1}, L_{q+2}, \dots, L_r]$ de la liste L seront fusionnés.

Exemple ci-après.

Programmation de la fusion.

Exemple.

- Entrée : la liste suivante avec $p = 2$, $q = 6$, $r = 10$

8 1 7 3 5 6 8 4 7 9 11 2 15

- En sortie :

8 1 7 3 4 5 6 7 8 9 11 2 15

Principe de programmation de la fusion.

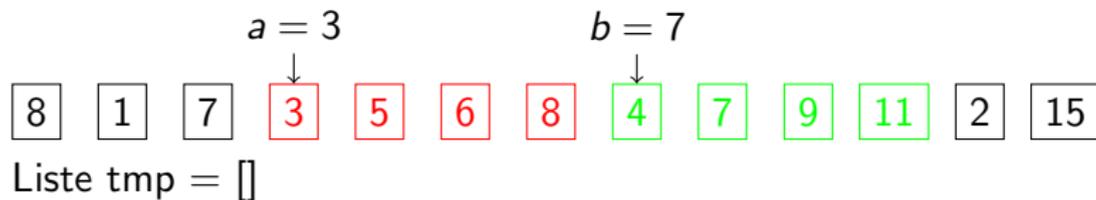
Pour fusionner les deux segments $[L_{p+1}, L_{p+2}, \dots, L_q]$ et $[L_{q+1}, L_{q+2}, \dots, L_r]$, nous allons utiliser :

- Une liste intermédiaire tmp qui sera l'analogie du paquet de cartes C dans l'explication initiale avec les paquets de cartes.
- Des indices a et b qui nous serviront à repérer notre avancement dans les segments à fusionner.

En dernière étape, il ne faudra pas oublier de recopier le contenu de la liste tmp à sa place dans la liste initiale L.

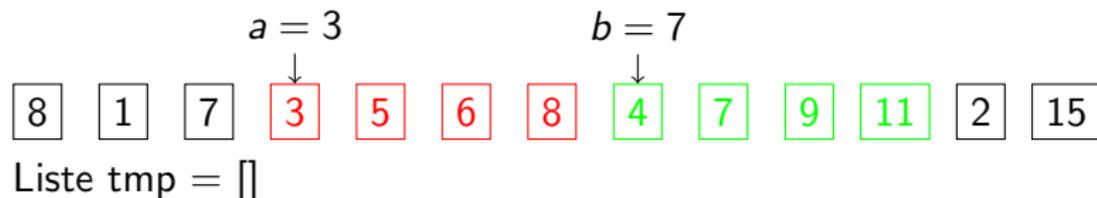
Fusion : explication du principe de programmation.

- Étape initiale.

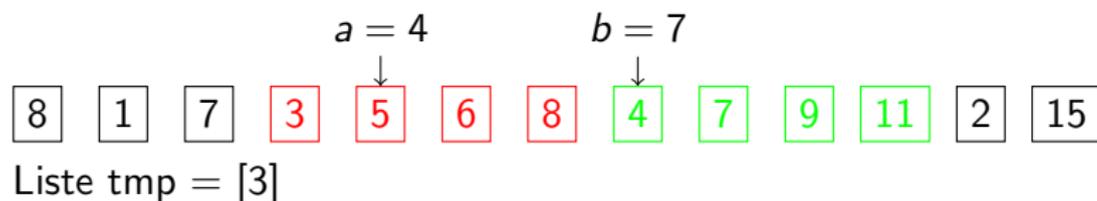


Fusion : explication du principe de programmation.

- Étape initiale.

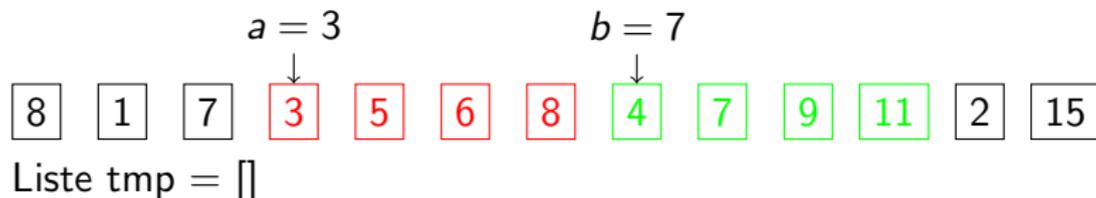


- Étape 1.

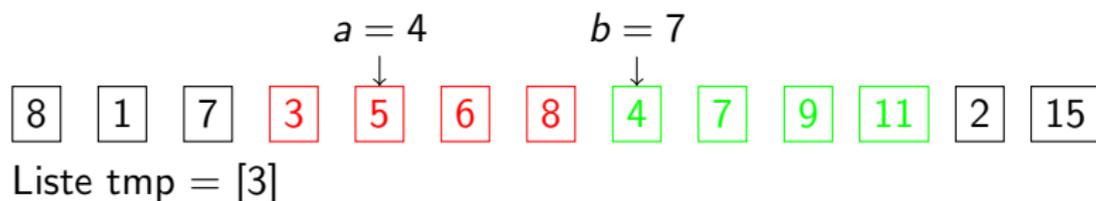


Fusion : explication du principe de programmation.

- Étape initiale.



- Étape 1.

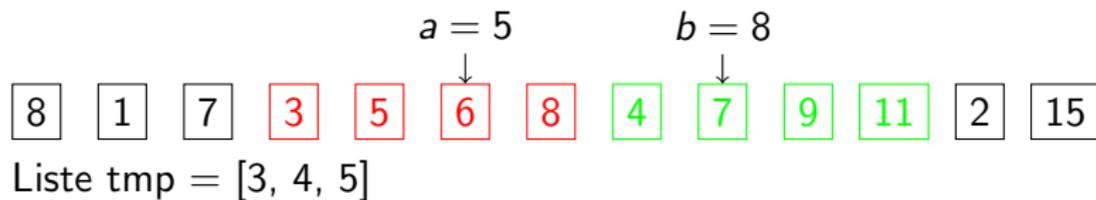


- Étape 2.



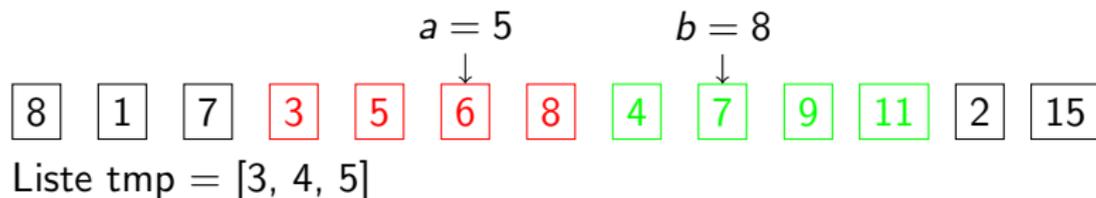
Fusion : explication du principe de programmation.

- Étape 3.

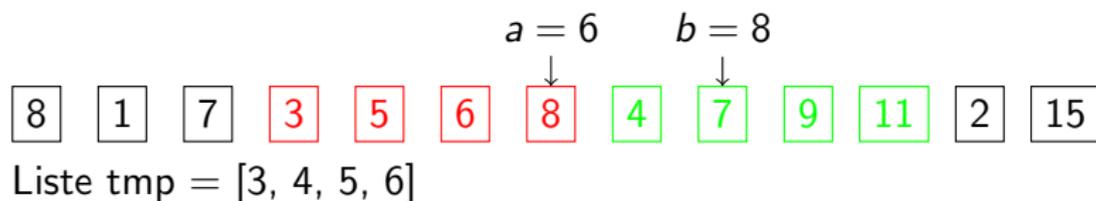


Fusion : explication du principe de programmation.

- Étape 3.

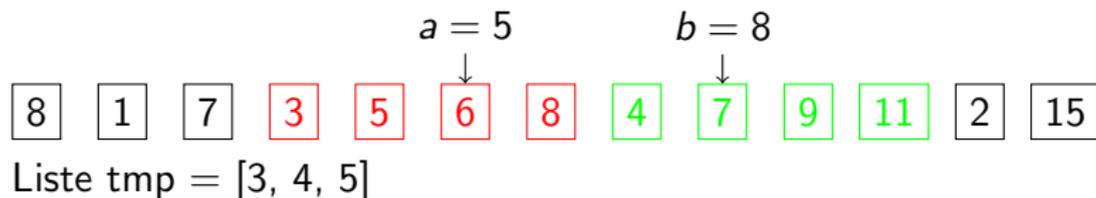


- Étape 4.

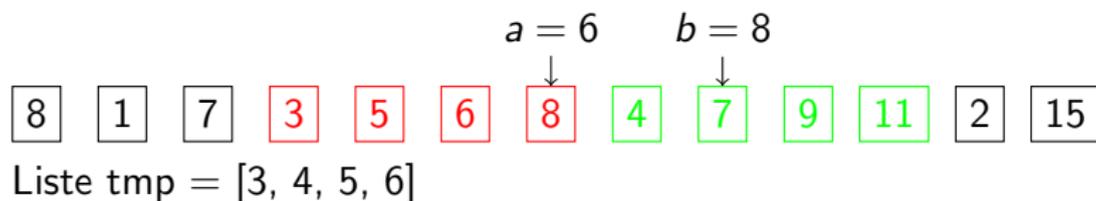


Fusion : explication du principe de programmation.

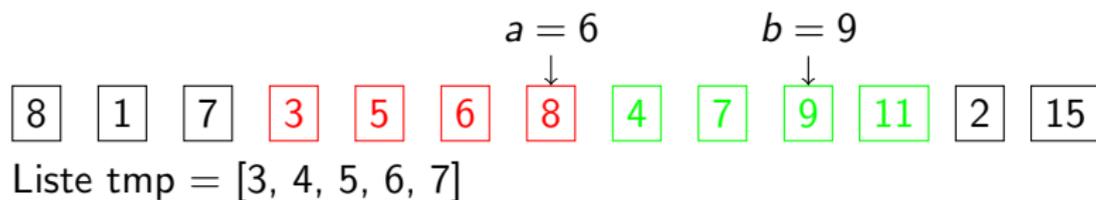
- Étape 3.



- Étape 4.

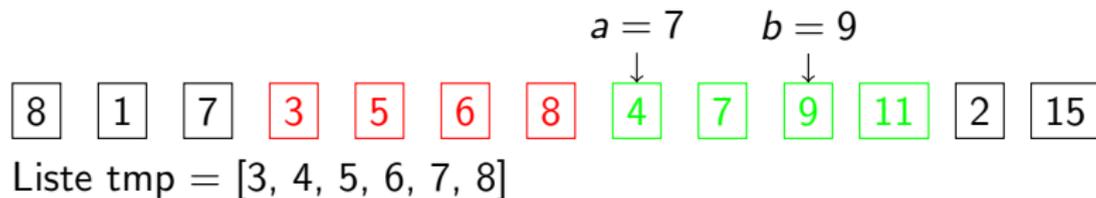


- Étape 5.

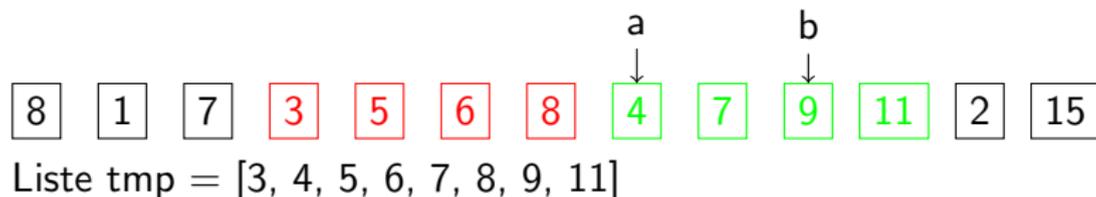


Fusion : explication du principe de programmation.

- Étape 6.



- Étape 7.



- Dernière étape : remplacer les segments rouge et vert de la liste initiale par la liste intermédiaire.

Exercice à rendre 2.

On propose ci-dessous une programmation de la procédure de fusion telle qu'elle a été décrite dans les pages précédentes.

Il manque une étape : la dernière étape qui consiste à recopier la liste temporaire tmp (nommée ci-dessous C) dans la liste L aux bonnes positions.

A vous d'écrire cette partie manquante.

Procédure fusion

Python

```
def fusion(p,q,r,L) :  
    C=[]  
    i,j=p+1,q+1  
    while i<=q and j<=r :  
        if L[i]<L[j] :  
            C.append(L[i])  
            i+=1  
        else :  
            C.append(L[j])  
            j+=1
```

Python

```
    if i==q+1 :  
        for k in range(j,r+1) :  
            C.append(L[k])  
    else :  
        for k in range(i,q+1) :  
            C.append(L[k])  
  
PARTIE À COMPLÉTER
```

du k -segmenté au $2k$ -segmenté.

On propose maintenant une programmation de la seconde procédure :



Python

```
def doublesegment(L,k) :  
    p=-1  
    while p+k<len(L) :  
        q=p+k  
        r=min(q+k,len(L)-1)  
        fusion(p,q,r,L)  
        p=r
```

Le tri par fusion.

Exercice à rendre numéro 3.

Écrire la procédure de tri par fusion, qui prend en entrée une liste et donne en sortie cette liste sous forme triée.

Le programme utilisera les fonctions précédentes.

Version récursive du tri par fusion

Vous avez à rendre (cf exercices 2 et 3 ci-dessus) une version itérative du tri par fusion.

On vous propose ci-dessous une version récursive de ce tri par fusion.
Cette version récursive est à **comprendre et à connaître**.

Version récursive : le principe.

- Input : liste L
- output : la liste L triée
- traitement.
 - Partager la liste en deux listes ListeGauche, ListeDroite (d'approximativement même longueur).
 - Trier ListeGauche et trier ListeDroite.
 - Fusionner ListeGauche et ListeDroite.

Version récursive : une programmation.

Python

```
def fusion(A,B) :  
    C=[]  
    while A and B :  
        if A[0]<B[0] : C.append(A.pop(0))  
        else : C.append(B.pop(0))  
    return C+A+B  
def trifusion(L) :  
    if len(L)<=1 : return L  
    return fusion(trifusion(L[:len(L)//2]),trifusion(L[len(L)//2:]))
```

Version récursive