

# Algorithmes de Tri

Radouan Dahbi

# Algorithme de tri

- Tri par sélection
- Tri à bulles
- Tri par insertion
- Tri rapide

# Définition

- Un **algorithme de tri** est un algorithme qui permet de réorganiser une liste d'éléments selon un ordre déterminé (croissant ou décroissant).
- Comprendre les algorithmes de tri est essentiel pour les bases de l'informatique car ils sont utilisés dans de nombreux domaines, tels que l'optimisation des recherches, la gestion des bases de données et bien plus encore.

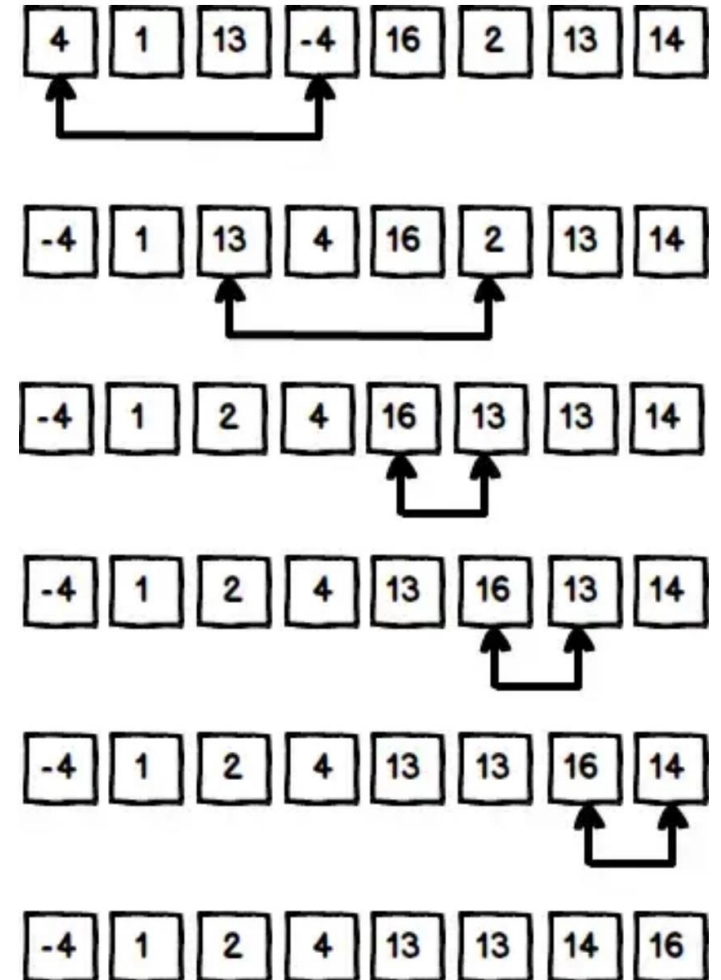
# Objectifs du cours

- Comprendre le fonctionnement des principaux algorithmes de tri
- S'exercer à les implémenter en Python

# Tri par sélection

## Principe :

- Rechercher le plus petit élément d'une liste et l'échanger avec le premier élément
- Rechercher le plus petit élément entre les positions 2 et  $n-1$  et l'échanger avec le deuxième élément
- ...



# Tri par sélection

## Pseudo-code :

fonction tri\_par\_selection(liste):

  n = longueur(liste)

  pour i de 0 à n-1:

    min\_indice = i

    min = liste[i]

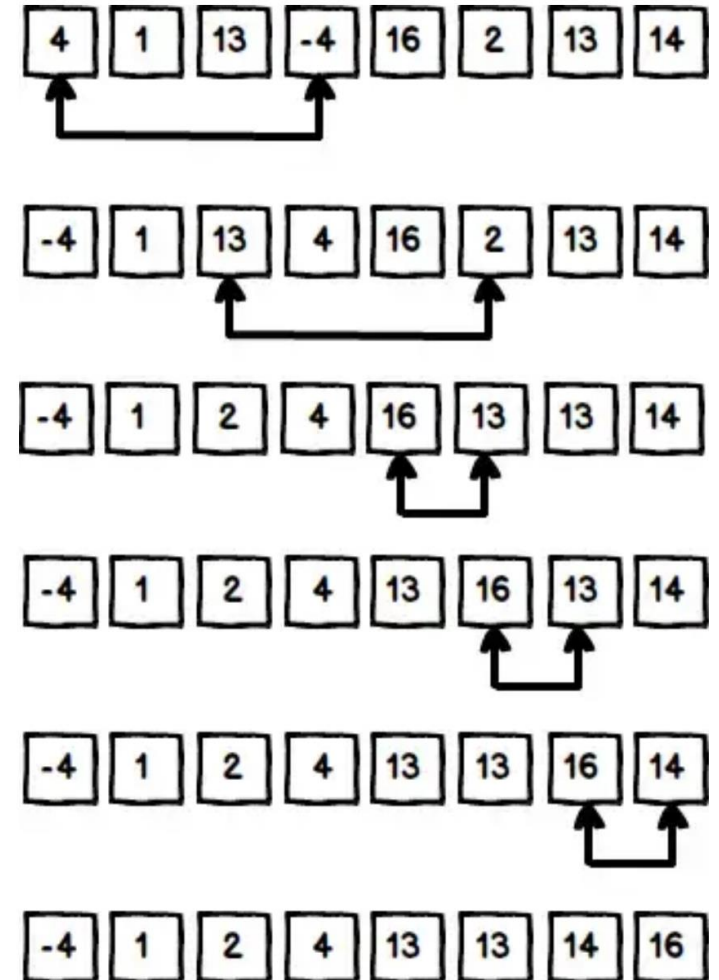
    pour j de i+1 à n:

      si liste[j] < min :

        min\_indice = j

        min = liste[j]

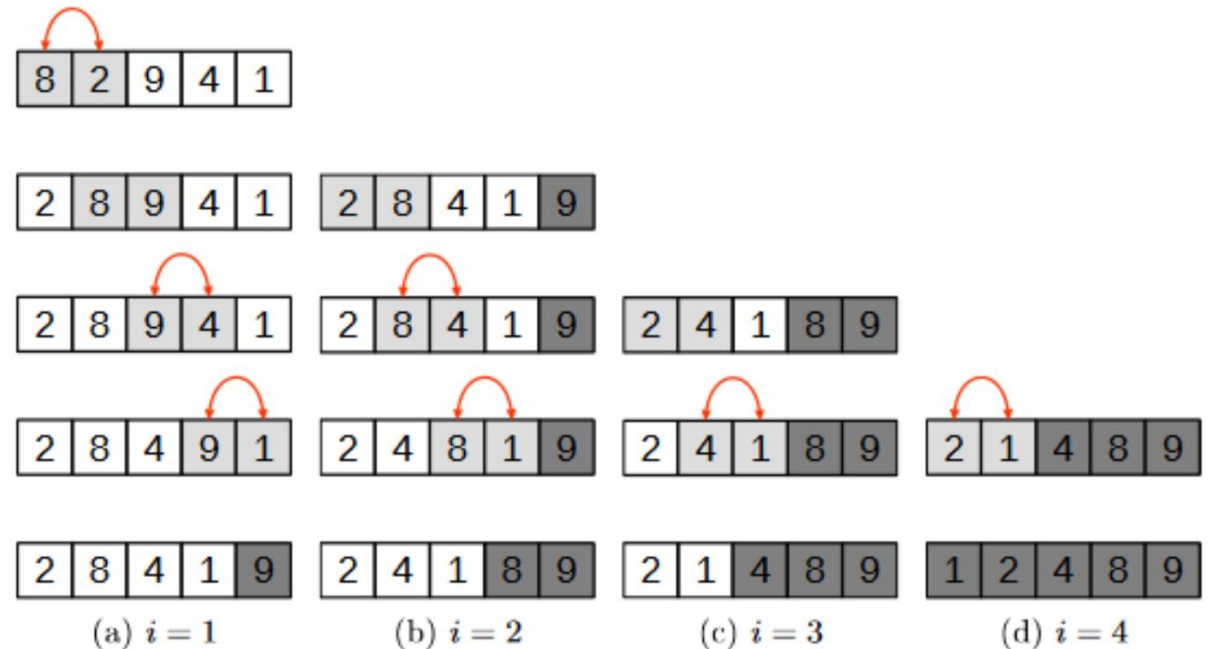
    échanger liste[i] et liste[min\_indice]



# Tri à bulles

## Principe :

- Comparer chaque élément d'une liste à son voisin et échanger leurs positions si nécessaire
- Répéter ce processus jusqu'à ce que la liste soit entièrement triée



# Tri à bulles

## Pseudo-code :

fonction tri\_a\_bulles(liste):

  n = longueur(liste)

  permut = vrai

  tant que permut == vrai

    permut = faux

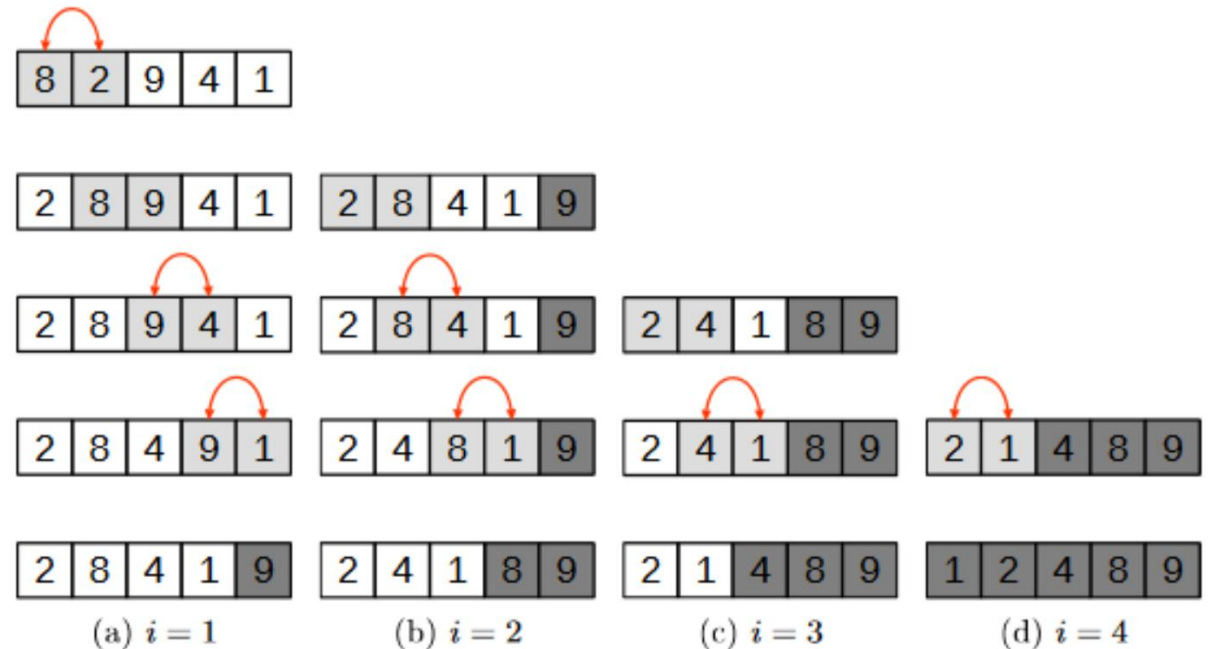
    pour i de 0 à n-1:

      si liste[i] > liste[i+1]:

        échanger liste[i] et liste[i+1]

        permut = true

  n=n-1





# Tri par insertion

## Principe :

Construire progressivement  
une partie triée de la liste en  
insérant chaque nouvel  
élément à la bonne position

9	2	7	1
---	---	---	---

2	9	7	1
---	---	---	---

2	7	9	1
---	---	---	---

1	2	7	9
---	---	---	---

# Tri par insertion

## Pseudo-code :

fonction tri\_par\_insertion(liste):

  n = longueur(liste)

  pour i de 1 à n:

    clé = liste[i]

    j = i - 1

    tant que j >= 0 et liste[j] > clé:

      liste[j+1] = liste[j]

      j = j - 1

    liste[j+1] = clé

9	2	7	1
---	---	---	---

2	9	7	1
---	---	---	---

2	7	9	1
---	---	---	---

1	2	7	9
---	---	---	---

# Tri rapide

## Principe :

Le tri rapide est un algorithme diviser-pour-régner. Il sélectionne un pivot, puis partitionne la liste de sorte que tous les éléments plus petits que le pivot se trouvent à sa gauche et tous ceux plus grands à sa droite. Ensuite, il trie récursivement les sous-listes.

5	9	7	3	8
---	---	---	---	---

5	3
---	---

7
---

9	8
---	---

5	3
---	---

7
---

9	8
---	---

3
---

5
---

7
---

8
---

9
---

3	5	7	8	9
---	---	---	---	---

# Tri rapide

## Principe :

```
fonction tri_rapide(liste):  
    si longueur(liste) <= 1:  
        retourner liste  
    sinon:  
        pivot = liste[0]  
        gauche = [x pour x dans  
liste[1:] si x <= pivot]  
        droite = [x pour x dans liste[1:]  
si x > pivot]  
        retourner tri_rapide(gauche) +  
[pivot] + tri_rapide(droite)
```

5	9	7	3	8
---	---	---	---	---

5	3
7	
9	8

5	3
7	
9	8

3	5	7	8	9
---	---	---	---	---

3	5	7	8	9
---	---	---	---	---