# -Cours-

# Les algorithmes des Tris

# Introduction

En informatique, un algorithme de tri est un algorithme qui permet d'organier une collection d'objets selon un ordre déterminé (Croissant ou décroissant).

Il existe plusieurs algorithmes des tris :

- **Le tri par sélection**
- **Le tri par bulles**
- **Le tri par insertion**
- **…**

# Tri par sélection

# Le tri par sélection

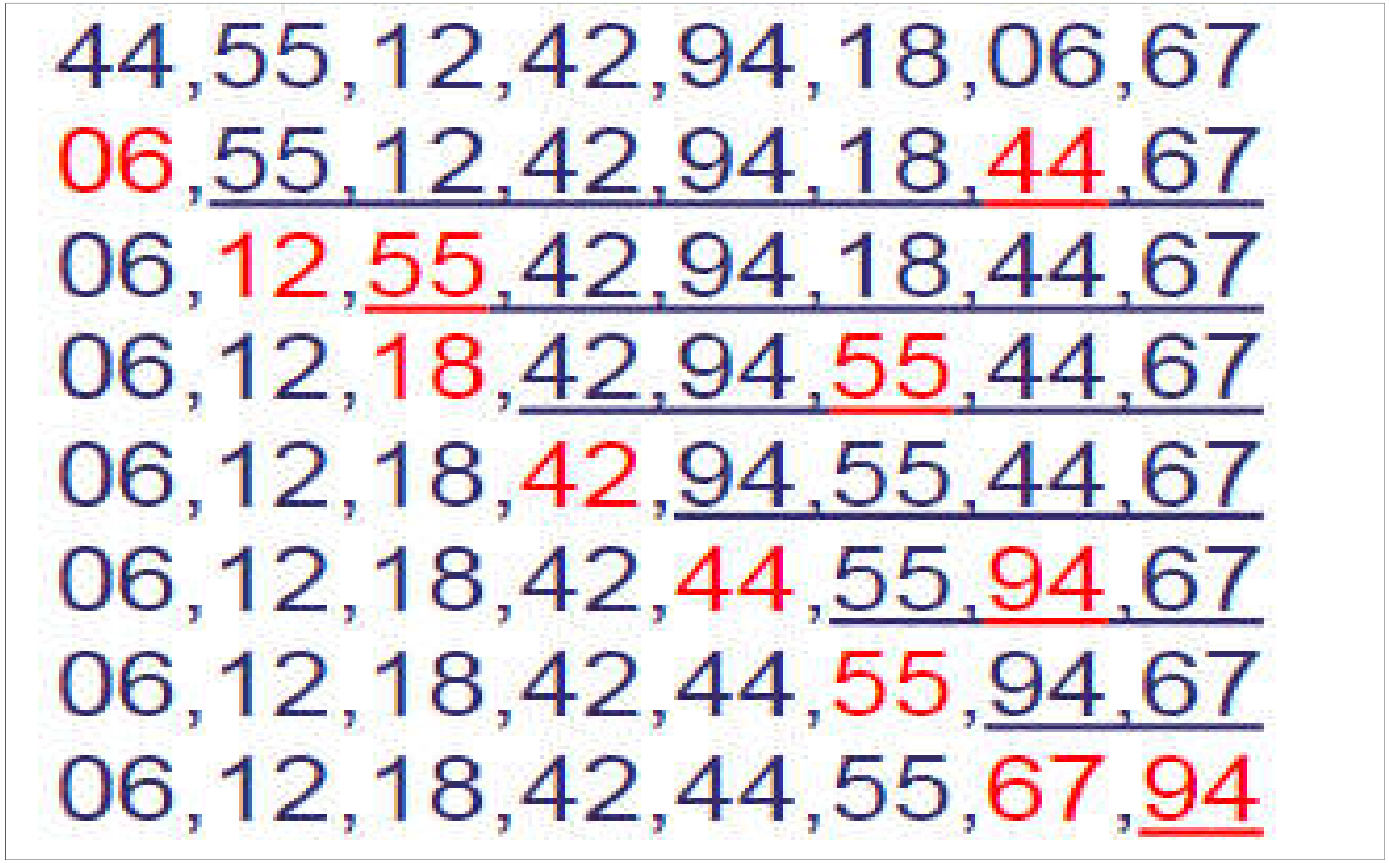
## ❖Le principe :

Le principe du **tri par sélection/échange** (ou tri par extraction) est d'aller **chercher le plus petit élément du tableau pour le mettre en premier**, puis de repartir du **second élément** et d'aller **chercher le plus petit élément du tableau pour le mettre en second**, etc...

Au $i^{ème}$ passage, on sélectionne donc l'élément ayant la plus petite valeur parmi les éléments **{T[i]...T[n-1]}** et **on l'échange avec T[i]**.

# Le tri par sélection

- Exemple de tri par sélection :



# Le tri par sélection

## ❖Programme de Tri par sélection:

```
def tri_selection(tab):
    n=len(tab) #taille de tableau
    for i in range(n-1):
            im=i #indice de l'element minimale
            for j in range(i+1,n):
                    if tab[j]<=tab[im]:
                            im=j #mémoriser l'indice du minimum
            #permutation
            tab[i],tab[im]=tab[im],tab[i]
```

# Tri par bulles

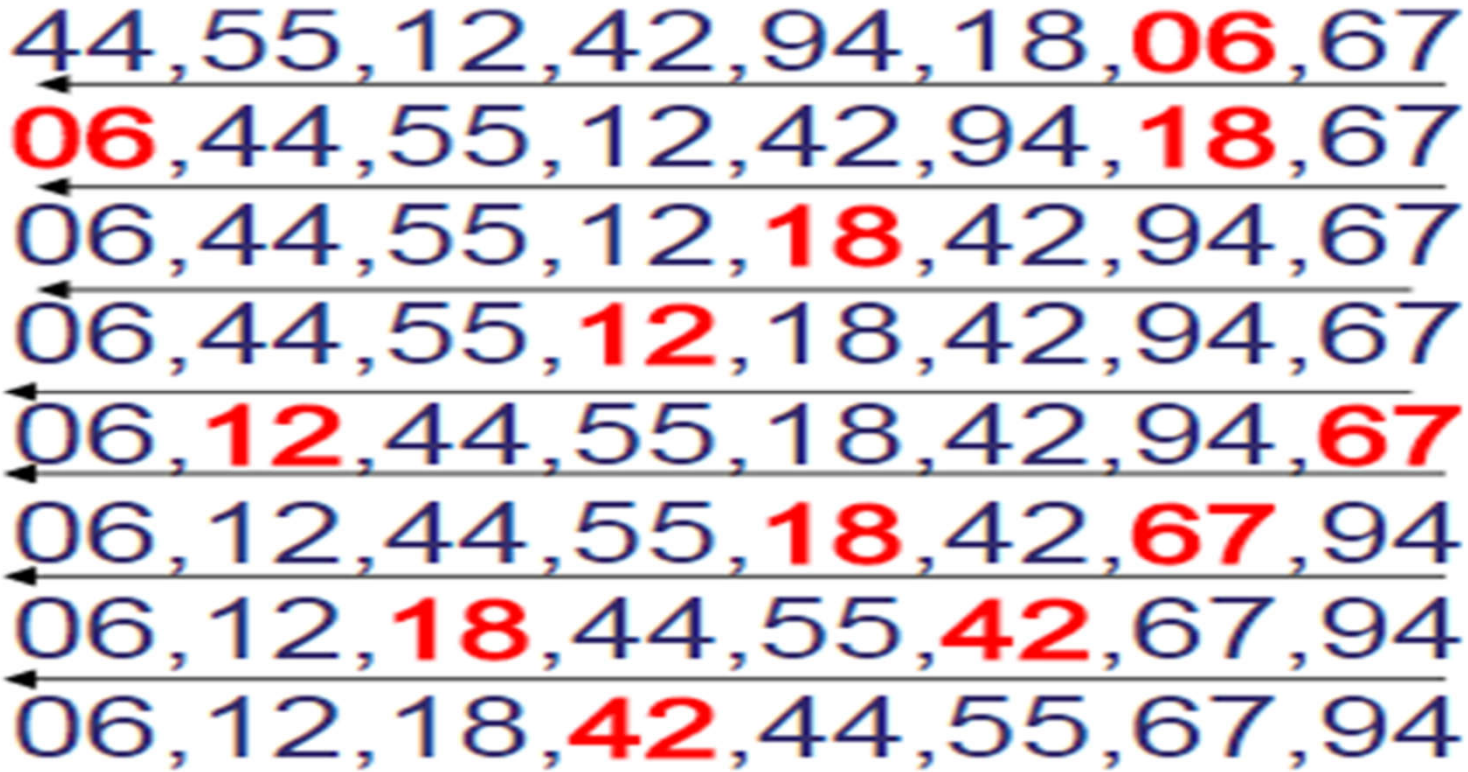
# Le tri par bulle

## ❖Le principe :

Le principe du **tri par bulle** est de **comparer deux à deux** les éléments e1 et e2 consécutifs d'un tableau et d'effecteur une **permutation si e1 > e2**. On continue de trier jusqu'à ce qu'il n'y ait plus de permutation

# Le tri par bulle

- Exemple de tri par bulle :



# Le tri par bulle

## Programme de Tri par bulle:

```
def tri_bulle(tab):
    n=len(tab) #taille de tableau
    permut=True #indice ou Flag de permutation
    while permut==True:
            permut=False #Flag pour interdire la permutation
            for i in range(n-1):
                    if tab[i]>tab[i+1]:
                            #faire la permutation
                            tab[i],tab[i+1]=tab[i+1],tab[i]
                            permut=True
```

# Le tri par bulle

- Programme de Tri par bulle:

```
def tri_bulle2(tab):
    n=len(tab) #taille de tableau
    for i in range(n):
            for j in range(n-1):
                    if tab[j]>tab[j+1]:
                            #faire la permutation
                            tab[j],tab[j+1]=tab[j+1],tab[j]
```

# Tri par insertion

# Le tri par insertion

## ❖Le principe :

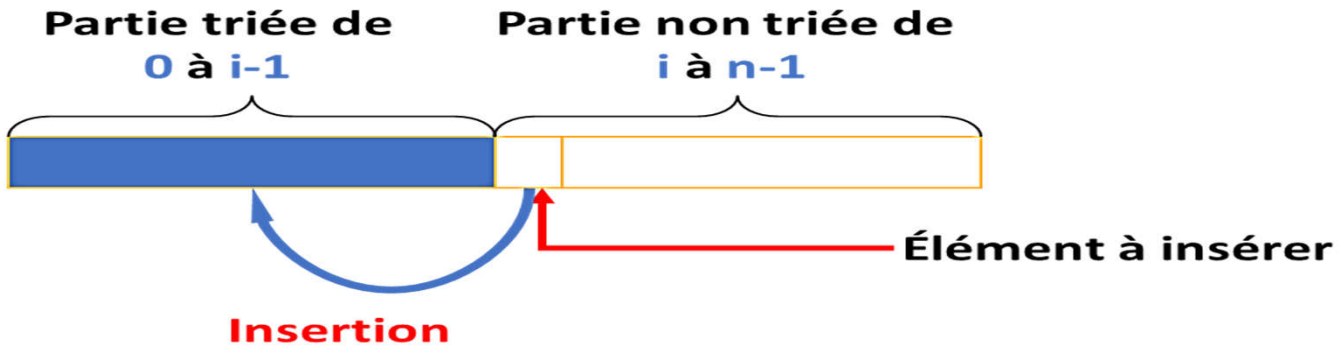
le principe est très simple : c'est l'algorithme qu'utilise naturellement l'être humain pour trier des obiets comme par exemple **des cartes à jouer**.

# Le tri par insertion

## ❖Le principe :

Le tri consiste pour chaque élément d'indice "i" à :

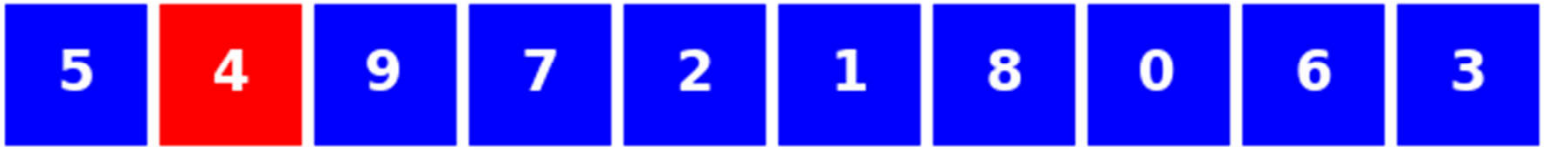
- Sélectionner l' $i^{ème}$ élément à insérer dans la partie non triée
- Insérer l'élément sélectionner dans la partie triée




***N.B: Initialement la partie triée contient le 1er élément***

# Le tri par insertion

- Exemple de tri par insertion:



# Le tri par insertion

## ❖Programme de Tri par insertion:

```
def tri_insertion(tab):
    n=len(tab)#taille de tableau
    for i in range(1,n):
            val=tab[i]#mémoriser la valeur de tab[i]
            pos=i #position
            while pos>0 and tab[pos-1]>val:
                    tab[pos]=tab[pos-1]#Affectation
                    pos=pos-1 #décrémenter la position pos par -1
            tab[pos]=val #Affecter val au position pos
```