Partie 4 : Tri itératif de tableau

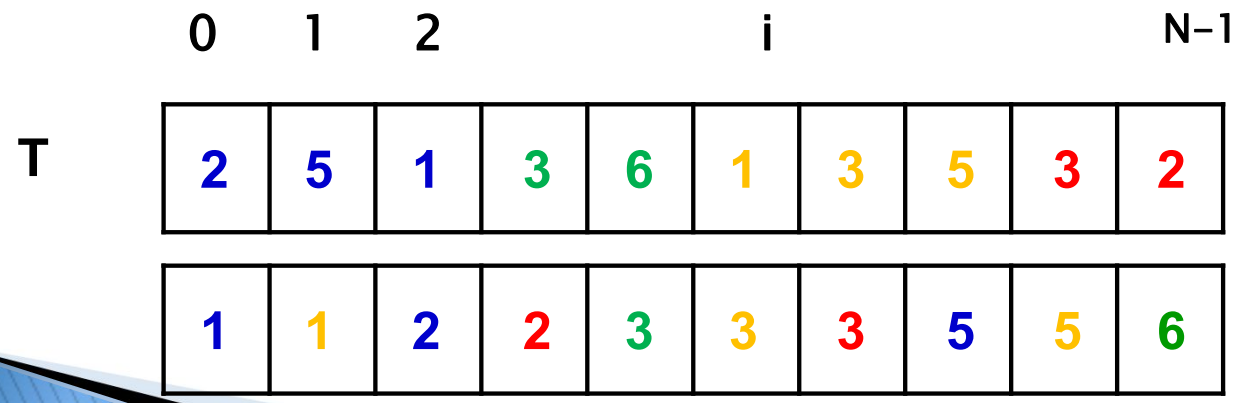
Hypothèses de travail :

- On manipule des entiers.
- On cherche à ordonner une séquences de N entiers.
- On considère que ces entiers sont dans un tableau.
- On considère que l'ordre recherché est l'ordre croissant.

Tri itératif de tableau

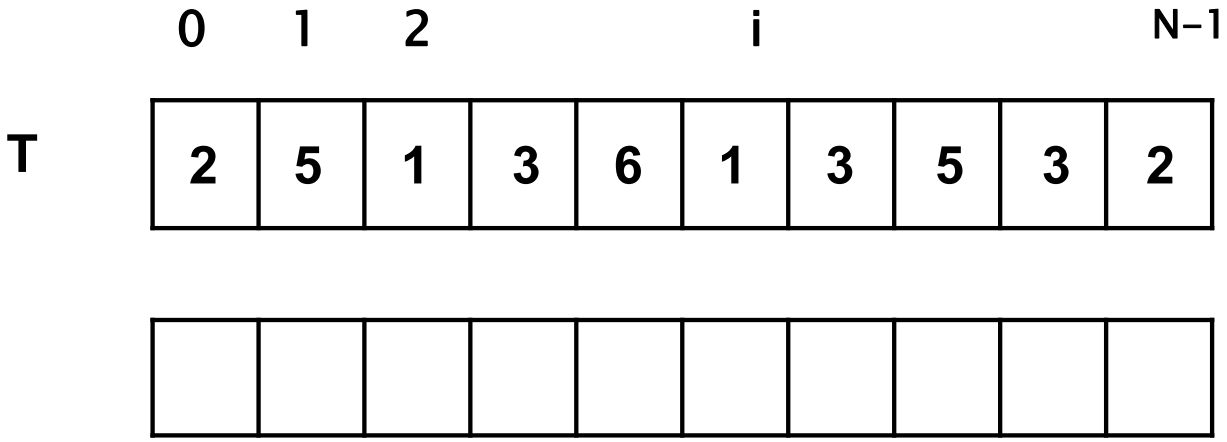
Critères d'analyses des algorithmes :

- Complexité en temps
- Complexité en mémoire
- Stabilité
  - Exemple :



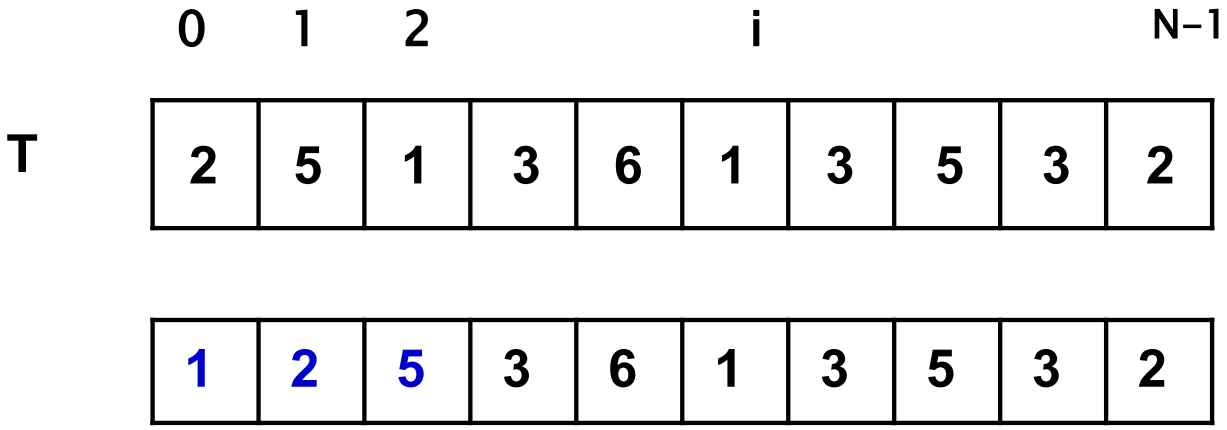
Tri itératif de tableau

Idées :



Tri itératif de tableau

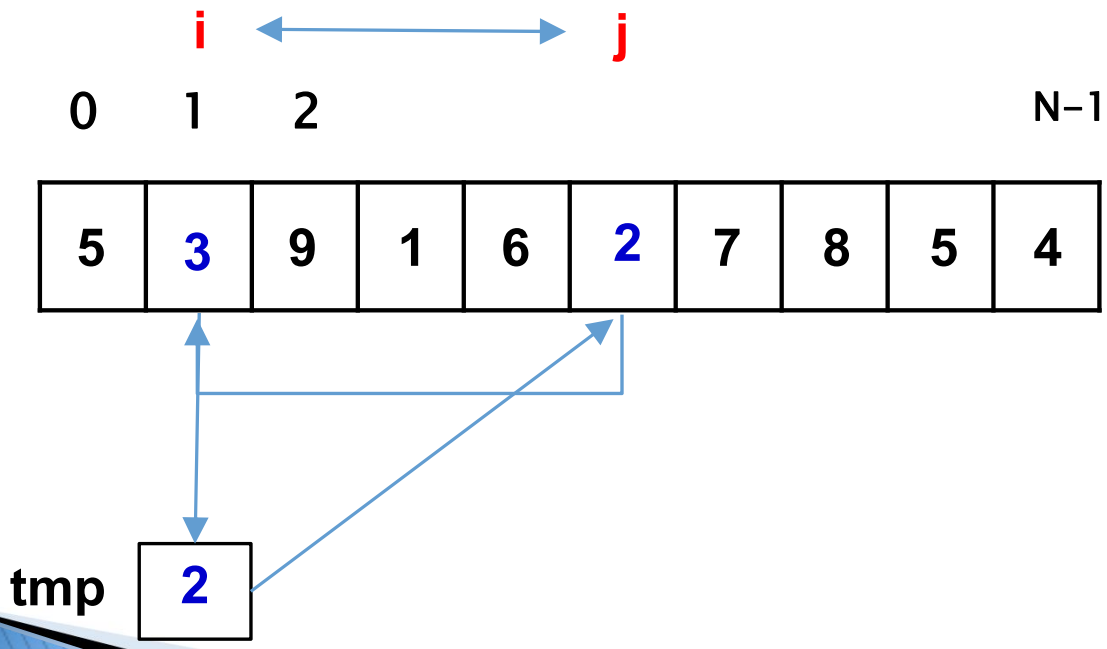
Idées :



Tri par sélection

Utilise la fonction

```
echanger (t, i, j)
```



Tri par sélection

Utilise la fonction

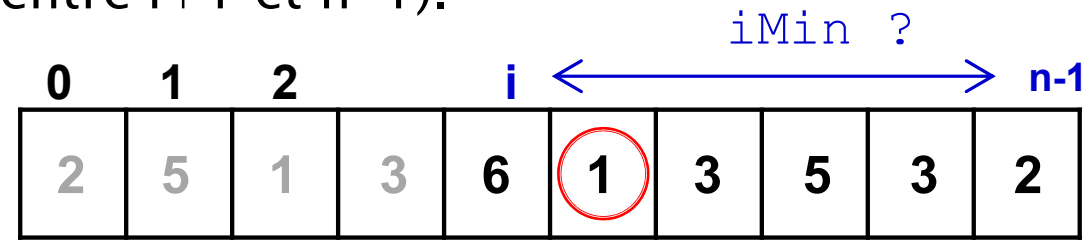
```
def echanger (t, i, j):
    tmp=t[i]
    t[i]=t[j]
    t[j]=tmp
```

Temps de calcul :

Tri par sélection

Basé sur la sélection du minimum :

- adapter l'algorithme pour l'utiliser sur un sous tableau (entre i+1 et n-1).

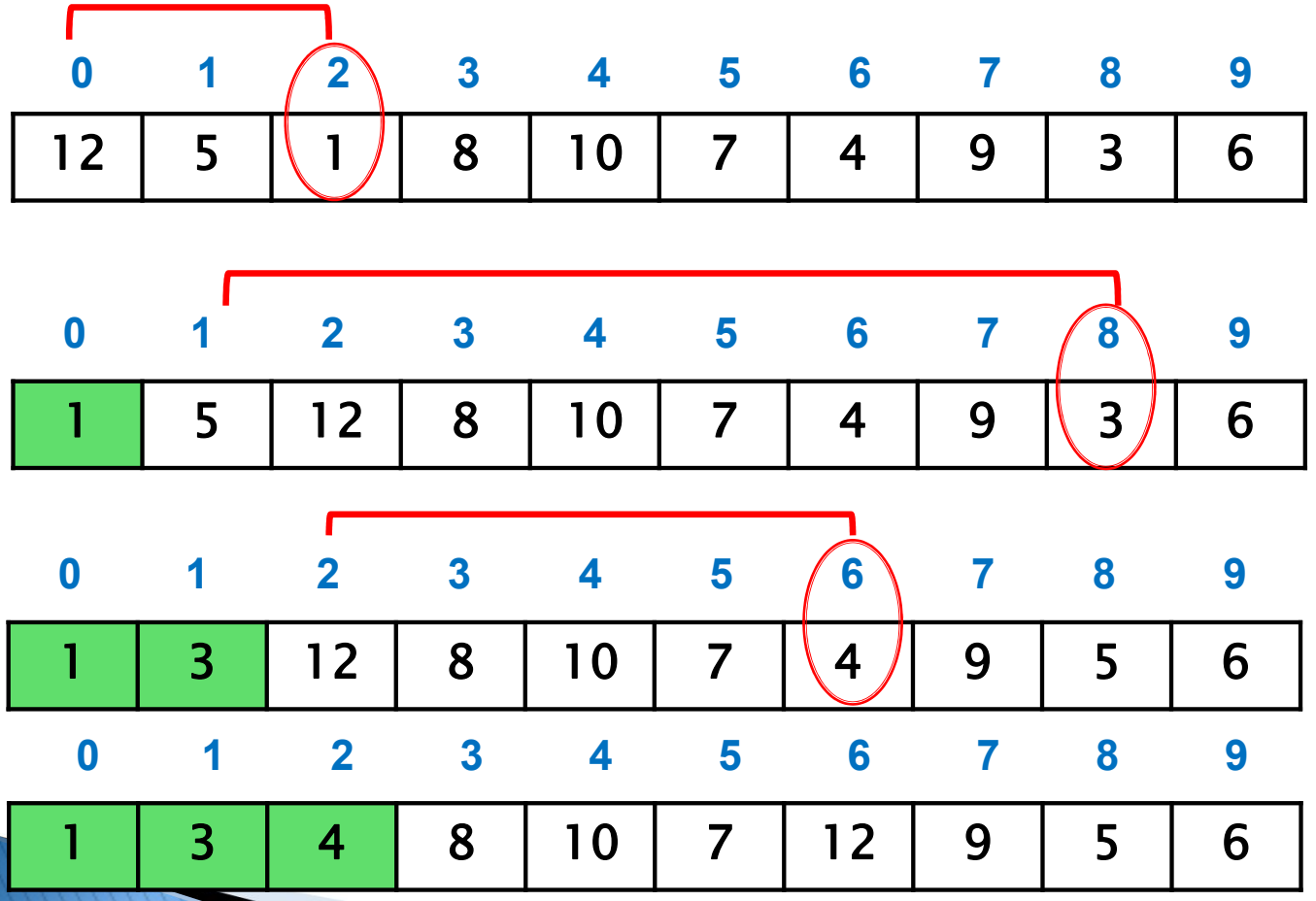



```
def indiceMin(t, n, i):
    iMin=i
    for j in range (i+1, n) :
        if(t[j]<t[iMin]):
            iMin=j
    return(iMin)
```

Tri par sélection

Exemple :
https://www.youtube.com/user/AlgoRythmics/videos

Tri par sélection



Tri par sélection

```
def triSelection (t,n):
  for  i in range(n) :
    iMin=i
    for j in range(i+1,n) :
      if(t[j]<t[iMin]):
        iMin=j
    if(i != iMin):
      echanger(t,i,iMin)
```

Complexité en temps ?
Complexité en mémoire ?
Stabilité ?

Proprieté :

Soit T un tableau d'entiers trié entre les indices i et j (i<=j).

Soit e un entier quelconque, alors on a l'une des propriétés suivantes :

- e≤T[i]
- il existe un unique entier k dans [i..j-1] tel que T[k]<e≤T[k+1]
- e>T[j]

On déduit de cette propriété deux algorithmes permettant de trier un tableau.

Proprieté :

0	1	2	3	4	5	6	7	8	9
1	3	4	8	10	7	12	9	5	6

Idées ?

Tri Insertion

C'est le tri des joueurs de cartes : insérer une nouvelle carte parmi celles déjà triées.

Hypothèse :

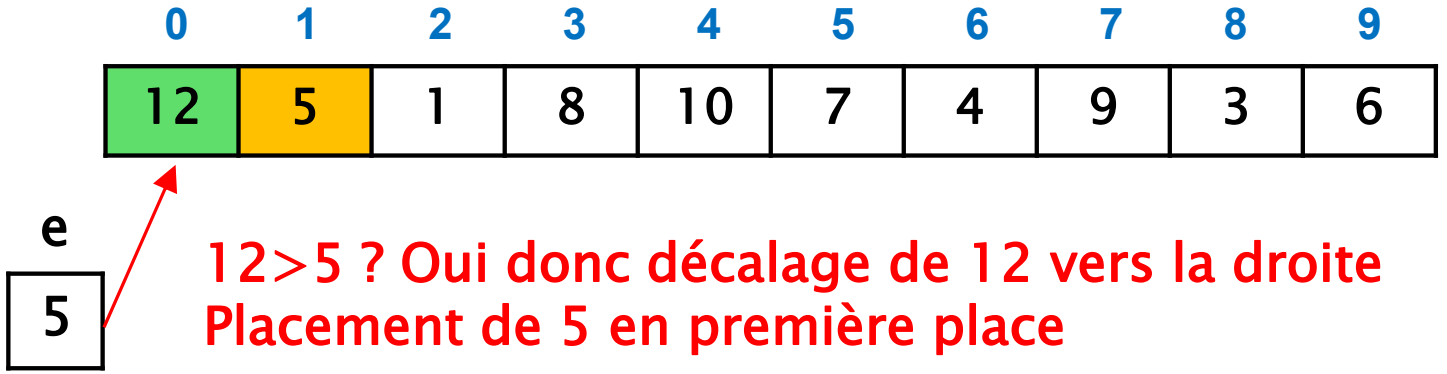
- Les éléments du tableau entre les indices 0 et i-1sont triés entre eux.
- Les éléments de t d'indice supérieur ou égal à i n'ont jamais été observé.

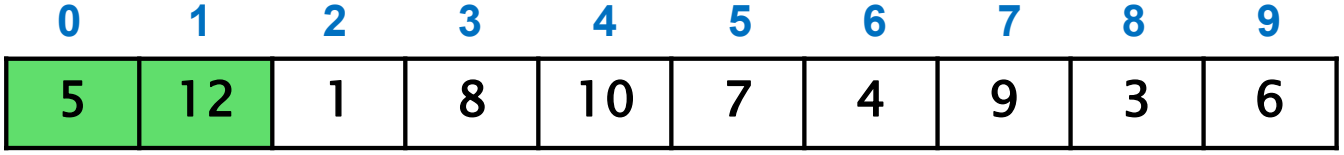
On doit mettre t[i] à sa place.

Tri insertion

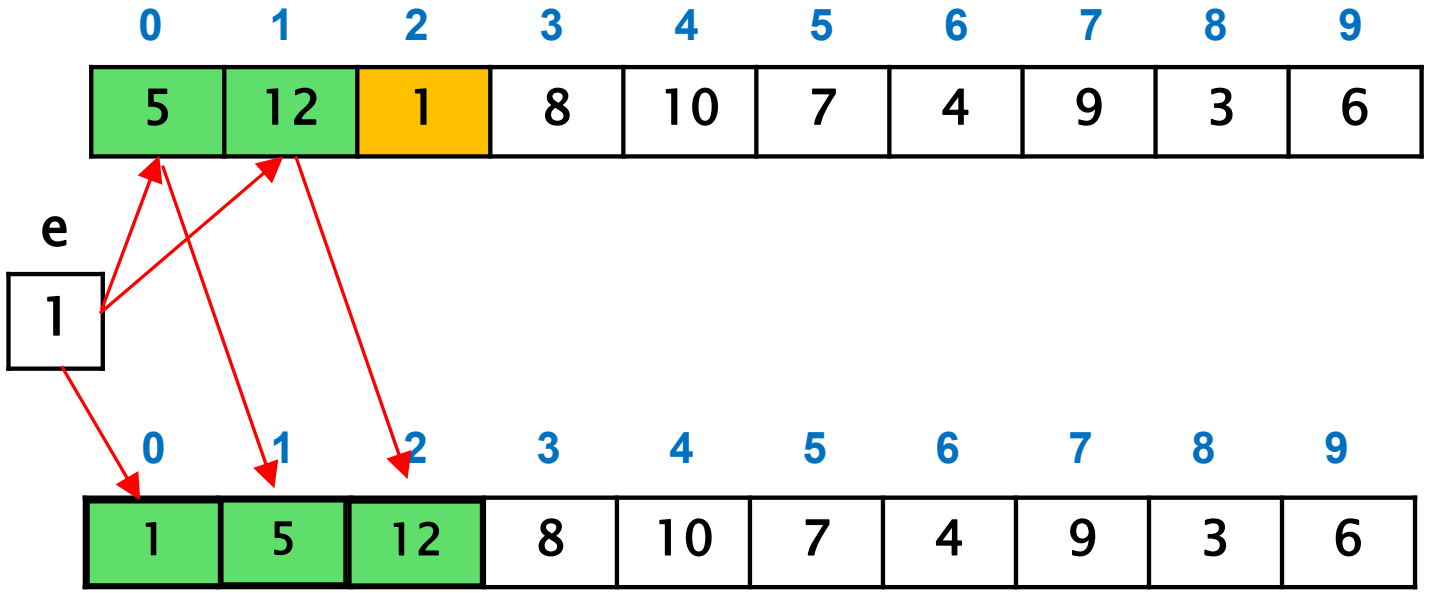
Exemple :
https://www.youtube.com/user/AlgoRythmics/videos

Tri insertion

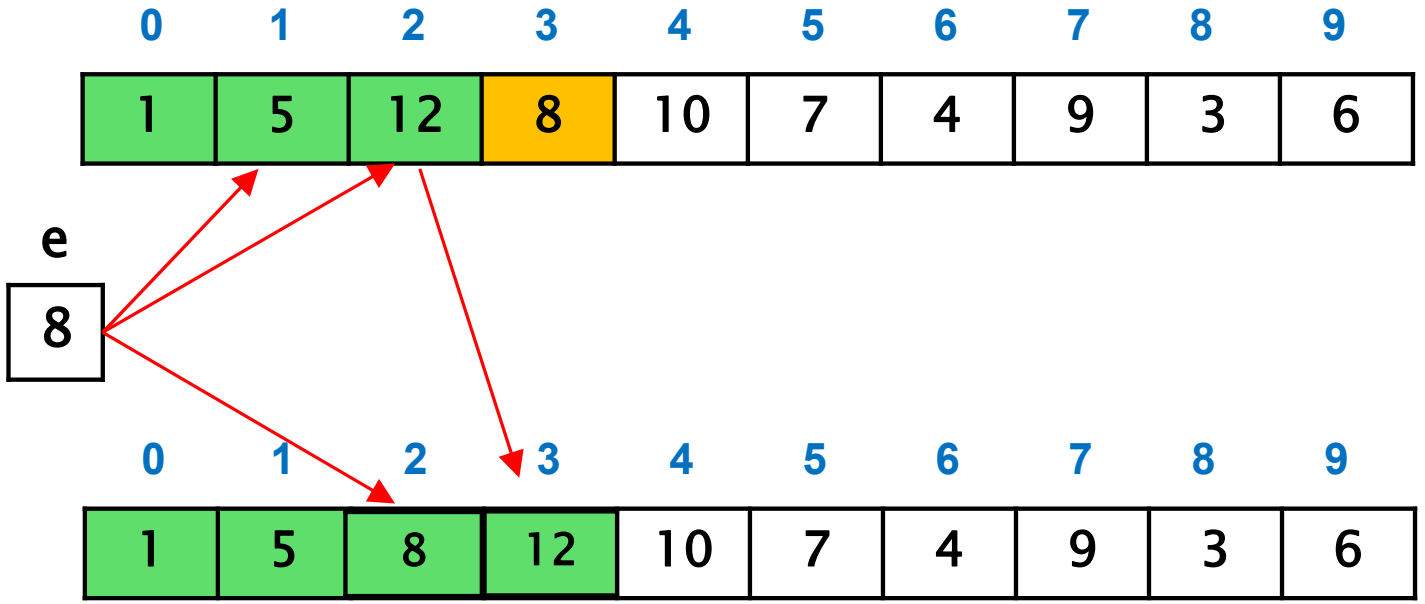




Tri insertion



Tri insertion



...

Tri Insertion

```
def triInsertion(t, n):
 for i in range(1,n):
  e=t[i]
  j=i-1
  while (j>=0 and t[j]>e):
    t[j+1]=t[j]
    j=j-1
  t[j+1]=e
```

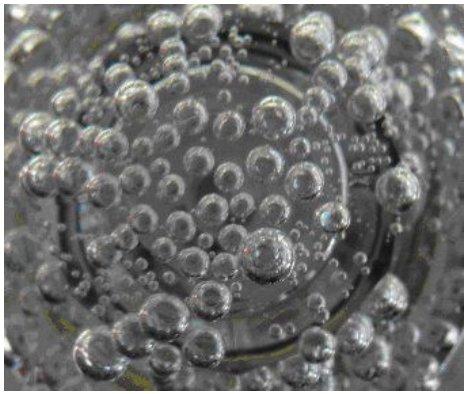
Complexité en temps au mieux? Au pire ?

Complexité en mémoire ?

Stabilité ?

Tri à bulles

Les bulles les plus grosses remontent en premier dans une boisson gazeuse.



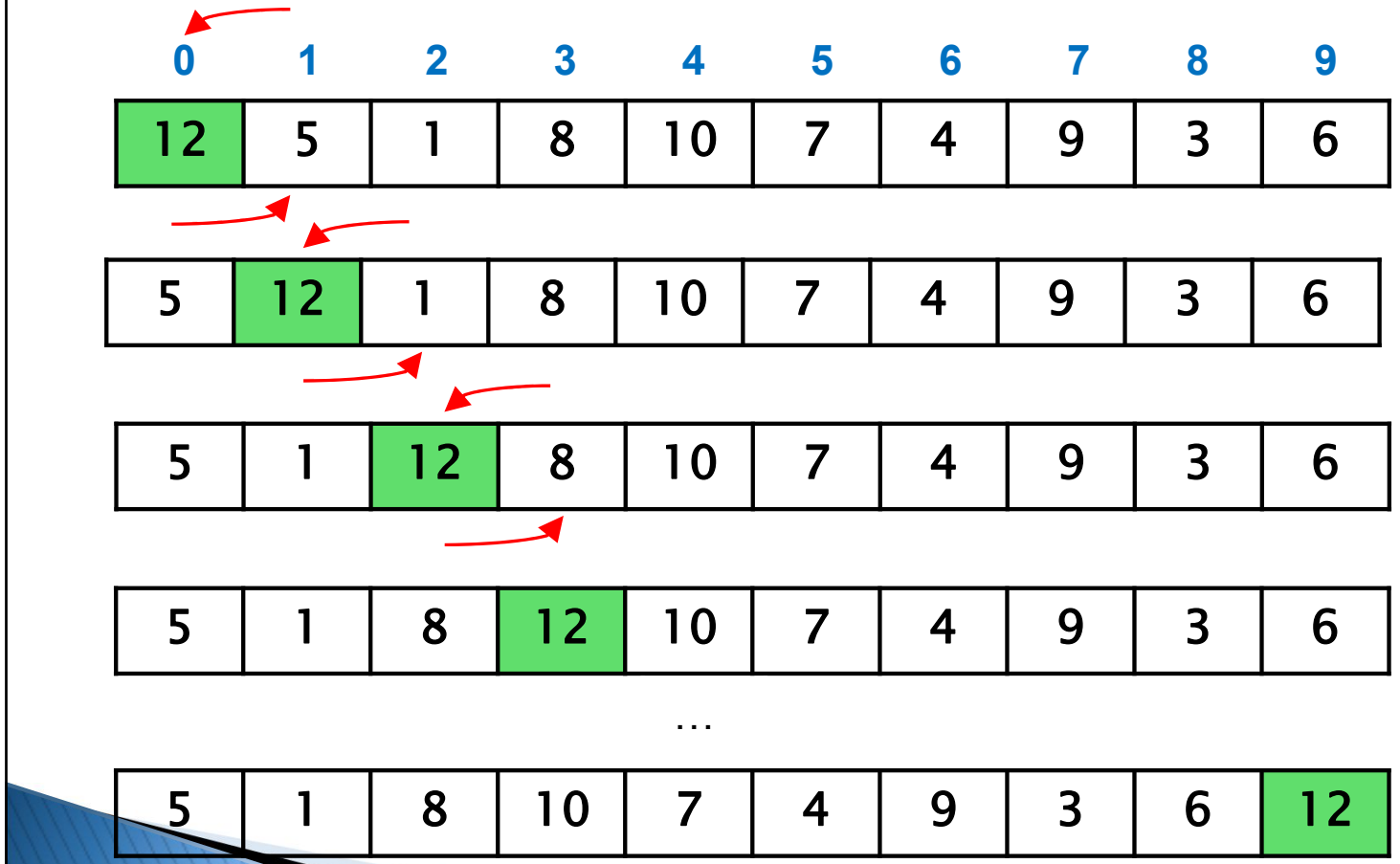
Hypothèse :

- Le tableau est trié entre les indices i et n-1.
- On échange deux à deux les éléments de t qui ne sont pas correctement ordonnés entre les indices 0 et i-1.

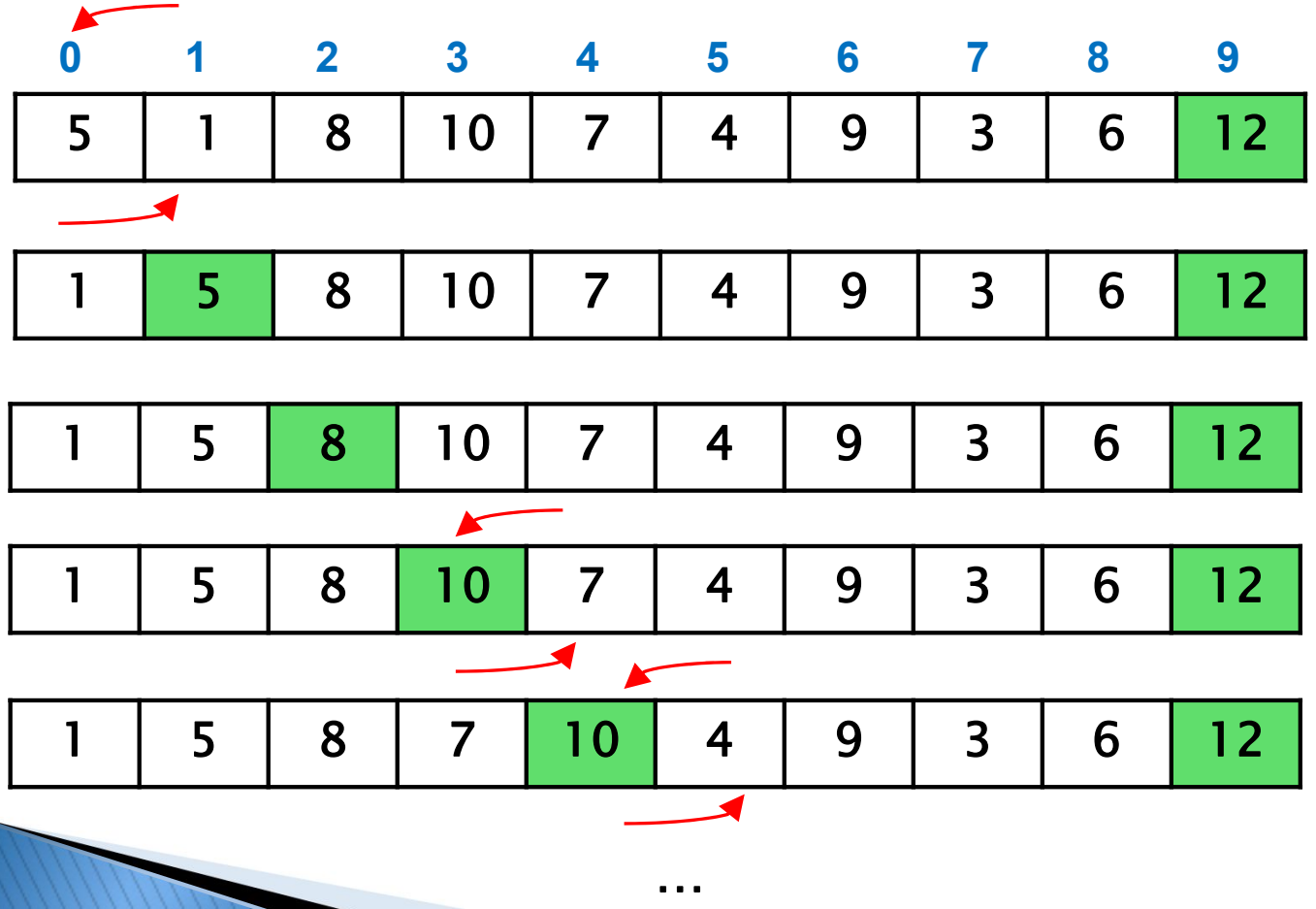
Tri à bulles

Exemple :
https://www.youtube.com/user/AlgoRythmics/videos

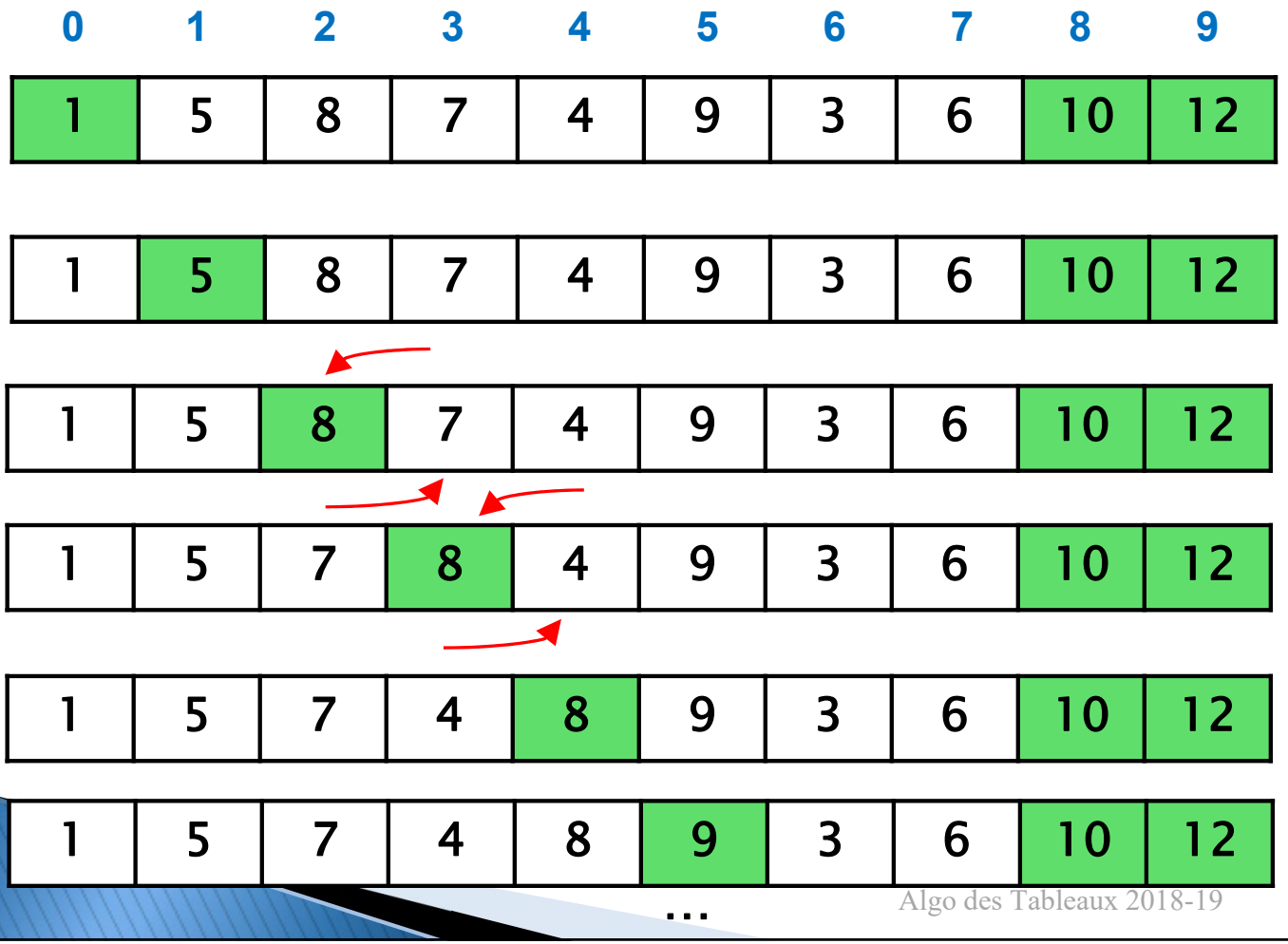
Tri à bulles


0 1 2 3 4 5 6 7 8 9
12 5 1 8 10 7 4 9 3 6
5 12 1 8 10 7 4 9 3 6
5 1 12 8 10 7 4 9 3 6
5 1 8 12 10 7 4 9 3 6
...
5 1 8 10 7 4 9 3 6 12

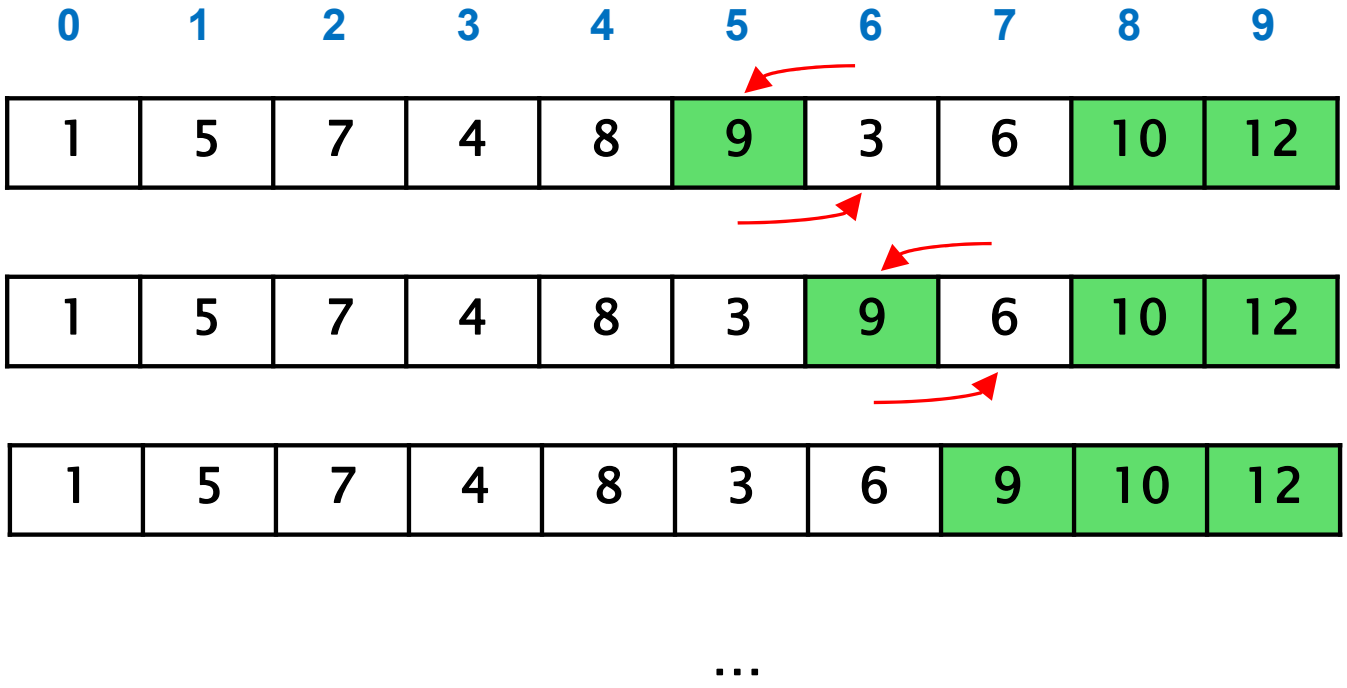
Tri à bulles


0 1 2 3 4 5 6 7 8 9
5 1 8 10 7 4 9 3 6 12
1 5 8 10 7 4 9 3 6 12
1 5 8 10 7 4 9 3 6 12
1 5 8 10 7 4 9 3 6 12
1 5 8 7 10 4 9 3 6 12
...

Tri à bulles


0 1 2 3 4 5 6 7 8 9
1 5 8 7 4 9 3 6 10 12
1 5 8 7 4 9 3 6 10 12
1 5 8 7 4 9 3 6 10 12
1 5 7 8 4 9 3 6 10 12
1 5 7 4 8 9 3 6 10 12
1 5 7 4 8 9 3 6 10 12
...

Tri à bulles



...

Question

- Trouver les 100 meilleurs notes en licence à l'université de Bordeaux en 2019 ?