

# NSI 1ère - Algorithmique - Tris 2

---

QK

## Seconde partie : pseudo code

---

# Tri par sélection

```
tri_selection(tableau t, entier n)
  pour i de 1 à n - 1
    min = i
    pour j de i + 1 à n
      si t[j] < t[min], alors min = j
    fin pour
    si min = i, alors échanger t[i] et t[min]
  fin pour
```

## Tri sélection à la main 1.

```
tableau = [1, 3, 4, 2]
```

```
Boucle (i allant de 1 à 3)
```

```
  i = 1
```

```
  min = 1
```

```
  Boucle (j allant de 2 à 4)
```

```
    j = 2 t[min] = 1 < t[2] = 3. Pas d'échange
```

```
    j = 3 t[min] = 1 < t[3] = 4. Pas d'échange
```

```
    j = 4 t[min] = 1 < t[4] = 2. Pas d'échange
```

```
  Fin boucle
```

```
rien à faire
```

## Tri sélection à la main 2.

`i = 2`

`min = 2`

Boucle (j allant de 3 à 4)

`j = 3 t[min] = 3 < t[3] = 4. Pas d'échange`

`j = 3 t[min] = 3 > t[4] = 2. On échange : min = 4`

Fin boucle

min a changé, on échange dans le tableau :

`tableau = [1, 2, 4, 3]`

## Tri sélection à la main 3.

i = 3

min = 3

Boucle (j allant de 4 à 4)

j = 4 t[min] = 4 > t[4] = 3. On échange : min = 4

Fin boucle

min a changé, on échange dans le tableau :

tableau = [1, 2, 3, 4]

# Le tri par insertion

On commence avec une liste déjà triée vide. On itère sur la liste et, à chaque tour on insère le premier élément non trié

à sa place dans la liste triée

Tri *stable* : il ne change pas l'ordre de deux éléments "égaux" Tri en *place* : il n'utilise pas plus de mémoire

## Exemple

Triés	Non Triés	Élément le plus à gauche
()	(1, 3, 4, 2)	(1)
(1)	(3, 4, 2)	(3)
(1, 3)	(4, 2)	(4)
(1, 3, 4)	(2)	(4)
(1, 2, 3, 4)		



## Example 2

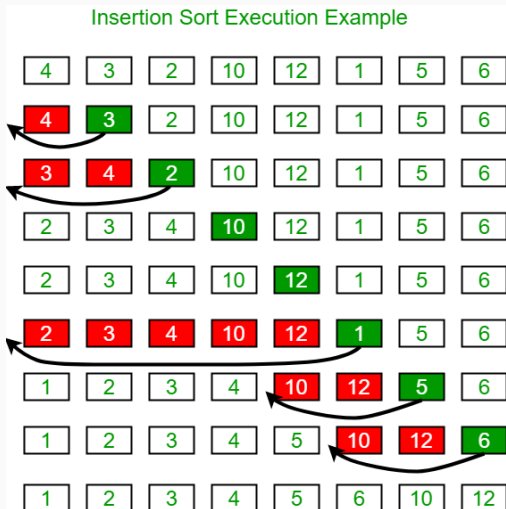


Figure 1: Tri par insertion

## Pseudo code

```
Tri Insertion(tableau t, entier n)
i = 1
Tant que i < n
    j = i
    Tant que j > 0 et t[j-1] > t[j]
        echanger t[j] et t[j-1]
        j = j - 1
    fin tant que
    i = i + 1
fin tant que
```

## Tri insertion à la main 1

```
tableau = [1, 3, 4, 2]
```

```
i = 1
```

```
Boucle tant que i < n:
```

```
    j = 1
```

```
    Boucle interne:
```

```
        t[0] = 1 < t[1] = 3-> sortie boucle
```

```
    i = 2
```

## Tri insertion à la main 2

2eme tour de la boucle externe

j = 2

Boucle interne

t[1] = 3 < t[2] = 4 -> sortie boucle

i = 3

## Tri insertion à la main 3

3eme tour de la boucle externe

j = 3

Boucle interne

t[2] = 4 > t[3] = 2 : on échange t[3] et t[2]

t = [1, 3, 2, 4]

j = 2

t[1] = 3 > t[2] = 2 : on échange t[1] et [2]

t = [1, 2, 3, 4]

j = 1

t[0] = 1 < t[1] = 2 -> sortie boucle

i = 4

Fin de la boucle externe (4 < 4 est Faux)