

# Cours 09 – Les tris

Novembre 2014

# Contrôle et Rattrapage TD

- Contrôle continu 2 : **19 novembre de 09h45 à 11h15**
- Groupe 2 MIASHS : TD **13 novembre 2014 de 09h45 à 13h00** en salle D101.

# Echange de deux éléments

```
void echange(int T[], int i, int j)
{
    int tmp;

    tmp = T[i];
    T[i] = T[j];
    T[j] = tmp;
}
```

# Inversion d'un tableau

- Echanger le premier élément avec le dernier élément du tableau
- Echanger le deuxième élément avec l'avant dernier
- ...
- Continuer jusqu'au milieu du tableau

# Inversion d'un tableau

8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

# Inversion d'un tableau

0							7
8	4	3	5	7	2	1	6

# Inversion d'un tableau

	1					6	
6	4	3	5	7	2	1	8

# Inversion d'un tableau

		2			5		
6	1	3	5	7	2	4	8



# Inversion d'un tableau

			3	4			
6	1	2	5	7	3	4	8

# Inversion d'un tableau

6	1	2	7	5	3	4	8
---	---	---	---	---	---	---	---

# Inversion d'un tableau

```
int T[N];  
int i;  
  
initialise(T);  
affiche_tableau(T);  
for( i=0; i<N/2; i++)  
{  
    echange(T,i,N-i-1);  
    affiche_tableau(T);  
}
```

# Les tris par comparaison

## Données

- Collection de *TailleMax* valeurs du même type rangées dans un tableau  $T$
- Un opérateur de comparaison  $\leq$

## But

Ré-ordonner les valeurs de  $T$  de telle sorte que :

$$T[i] \leq T[i + 1], \forall i \in \{0 \dots \textit{TailleMax} - 2\}$$

# Quelques algorithmes de tris

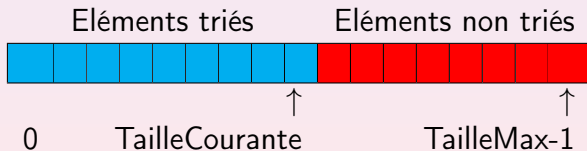
- Le tri par insertion
- Le tri par sélection
- Le tri à bulles (par permutation)

# Le tri par insertion

## Principe Général

A tout moment le tableau  $T$  est séparé en 2 parties :

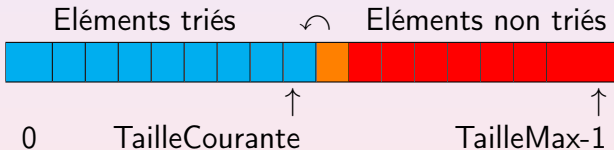
- $T[0] \dots T[TailleCourante]$  : Partie déjà triée du tableau
- $T[TailleCourante + 1] \dots T[TailleMax - 1]$  : Partie non triée du tableau.



# Le tri par insertion (2)

## Une Etape

- Prendre un élément non encore trié ;
- L'insérer à sa place dans l'ensemble des éléments triés.



## Le tri par insertion (3)

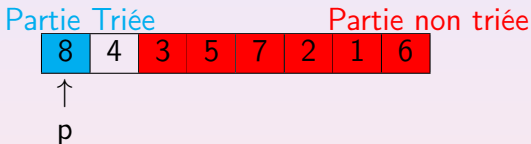
```
void TriInsertion(int T[], int N)
{
    int TC, i, p, tmp;
    for (TC = 0; TC < N-1; TC++)
    {
        tmp = T[TC+1];
        p = 0;
        while( T[p] < tmp)
            p = p + 1;
        for (i = TC; i >= p; i--)
            T[i+1] = T[i];
        T[p] = tmp;
    }
}
```



# Un exemple de tri par insertion

8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

# Un exemple de tri par insertion



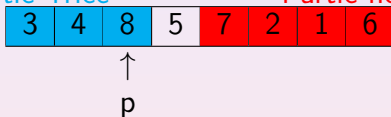
# Un exemple de tri par insertion



# Un exemple de tri par insertion

Partie Triée

Partie non triée



# Un exemple de tri par insertion

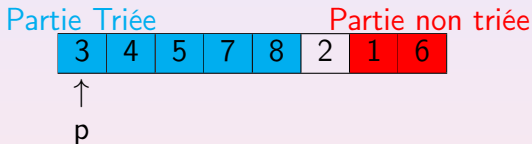
Partie Triée

Partie non triée

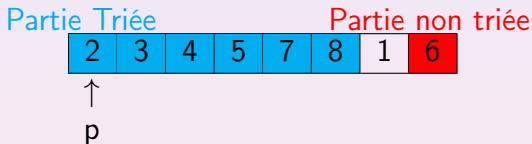


p

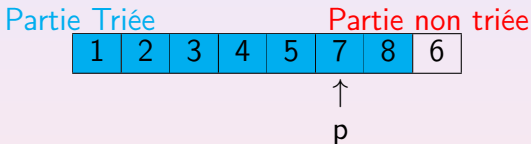
# Un exemple de tri par insertion



# Un exemple de tri par insertion



# Un exemple de tri par insertion





# Un exemple de tri par insertion

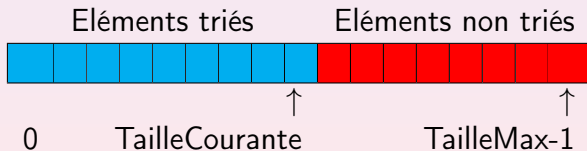
1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

# Le tri par sélection

## Principe Général

A tout moment le tableau  $T$  est séparé en 2 parties :

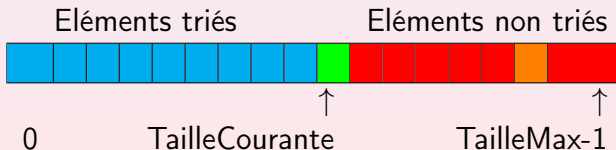
- $T[0] \dots T[TailleCourante]$  : Partie déjà triée du tableau
- $T[TailleCourante + 1] \dots T[TailleMax - 1]$  : Partie non triée du tableau.



# Le tri par sélection (2)

## Une Etape

- Prendre l'élément minimum dans la partie non triée.
- L'insérer à la fin de la partie triée.



## Le tri par sélection (3)

```
int RechercheIndMin(int T[], int N, int d)
{
    int indmin;
    int i;

    indmin = d;
    for(i=d; i<N; i++)
    {
        if (T[i] < T[indmin])
            indmin = i;
    }
    return indmin;
}
```

## Le tri par sélection (3)

```
void TriSelection(int T[], int N)
{
    int TC, indmin;
    for (TC = 0; TC < N-1; TC++)
    {
        indmin = RechercheIndMin(T,N,TC);
        exchange(T,TC,indmin);
    }
}
```

# Un exemple de tri par sélection

8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	4	3	5	7	2	8	6
---	---	---	---	---	---	---	---



# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	2	3	5	7	4	8	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	2	3	5	7	4	8	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	2	3	4	7	5	8	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	2	3	4	5	7	8	6
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

Partie non triée

1	2	3	4	5	6	8	7
---	---	---	---	---	---	---	---

# Un exemple de tri par sélection

Partie Triée

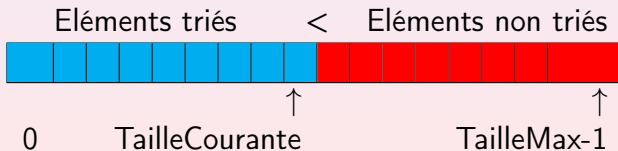
Partie non triée

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

# Tri par permutation

## Principe général

- Si deux éléments voisins ne sont pas ordonnés correctement, on les échange.
- Deux parties dans le tableau :
  - Une partie avec des éléments triés
  - Une partie avec des éléments non triésde telle sorte que les éléments de la partie triée sont inférieurs aux éléments de la partie non triée.



## Tri par permutation (2)

```
void Bulles(int T[], int N)
{
    int TC,i,tmp;

    for(TC=1;TC<N;TC++)
    {
        for(i=N-1;i>=TC;i--)
        {
            if (T[i-1] > T[i])
                exchange(T,i,i-1);
        }
    }
}
```



8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

TC



8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

TC



8	4	3	5	7	2	1	6
---	---	---	---	---	---	---	---

TC



8	4	3	5	7	1	2	6
---	---	---	---	---	---	---	---

TC



8	4	3	5	1	7	2	6
---	---	---	---	---	---	---	---

TC



8	4	3	1	5	7	2	6
---	---	---	---	---	---	---	---

TC



8	4	1	3	5	7	2	6
---	---	---	---	---	---	---	---

TC



8	1	4	3	5	7	2	6
---	---	---	---	---	---	---	---



TC



1	8	4	3	5	7	2	6
---	---	---	---	---	---	---	---

TC



1	8	4	3	5	7	2	6
---	---	---	---	---	---	---	---

TC



1	8	4	3	5	2	7	6
---	---	---	---	---	---	---	---

TC



1	8	4	3	2	5	7	6
---	---	---	---	---	---	---	---

TC



1	8	4	2	3	5	7	6
---	---	---	---	---	---	---	---

TC

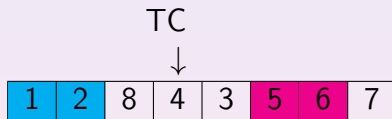


1	8	2	4	3	5	7	6
---	---	---	---	---	---	---	---

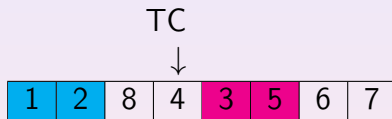
TC

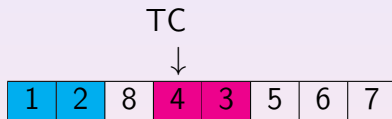


1	2	8	4	3	5	7	6
---	---	---	---	---	---	---	---









TC



1	2	8	3	4	5	6	7
---	---	---	---	---	---	---	---

TC



1	2	3	8	4	5	6	7
---	---	---	---	---	---	---	---

TC



1	2	3	8	4	5	6	7
---	---	---	---	---	---	---	---

TC



1	2	3	8	4	5	6	7
---	---	---	---	---	---	---	---

TC



1	2	3	8	4	5	6	7
---	---	---	---	---	---	---	---

TC



1	2	3	4	8	5	6	7
---	---	---	---	---	---	---	---



