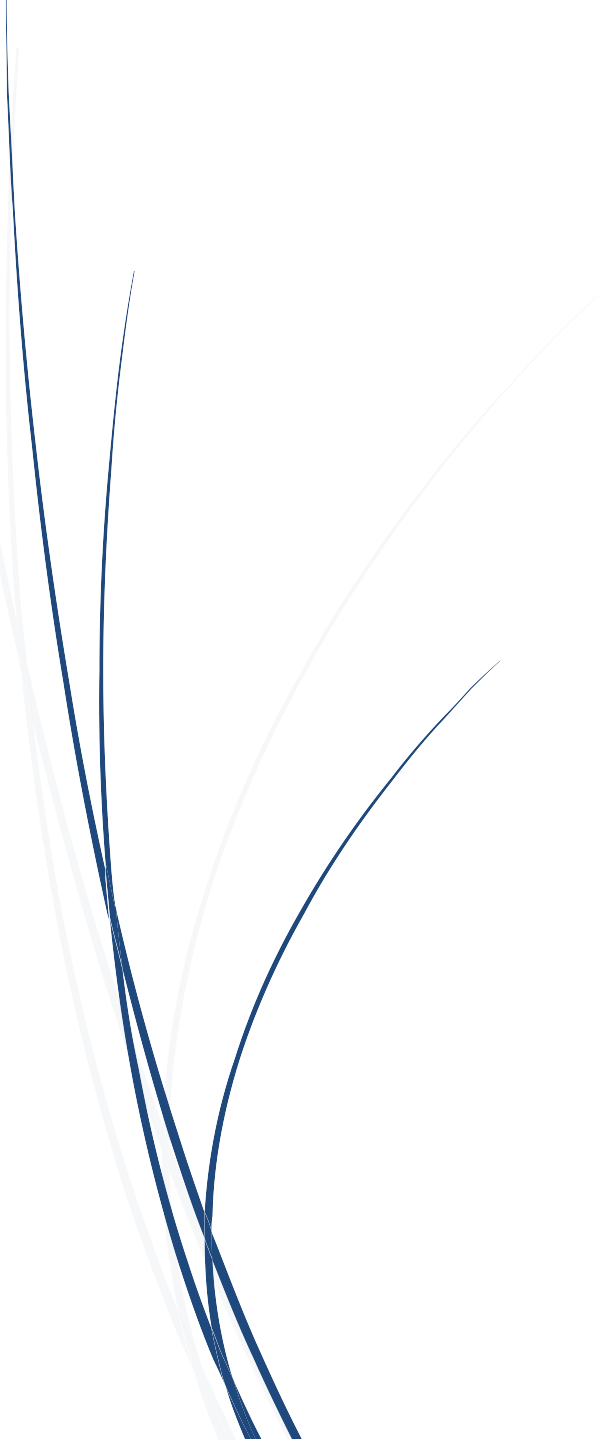


Les Tri

Trier les données



Organiser une collection d'objets selon une relation d’ordre déterminée.

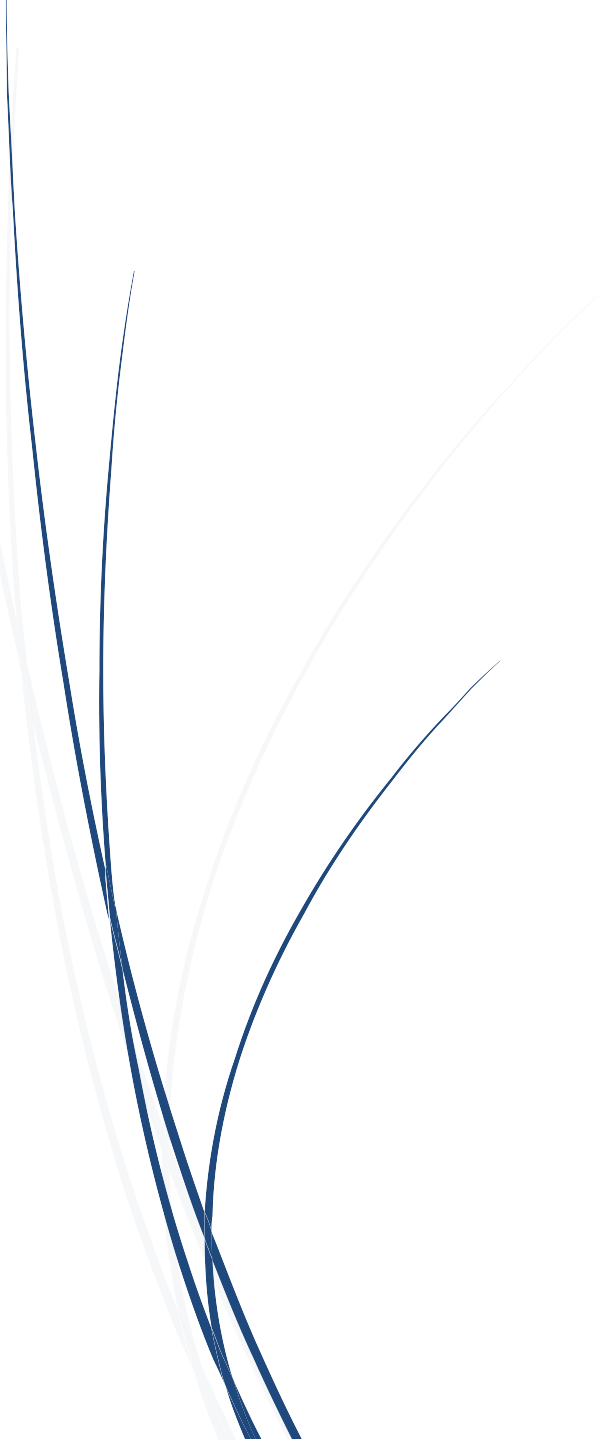
Exemple:

* Ordre croissant, décroissant des valeurs numériques,
* Ordre croissant des mots, …

Pourquoi trier?

* Faciliter la recherche (recherche dichotomique)
* Faciliter la gestion en générale

Exemple



## Trier en ordre croisant:

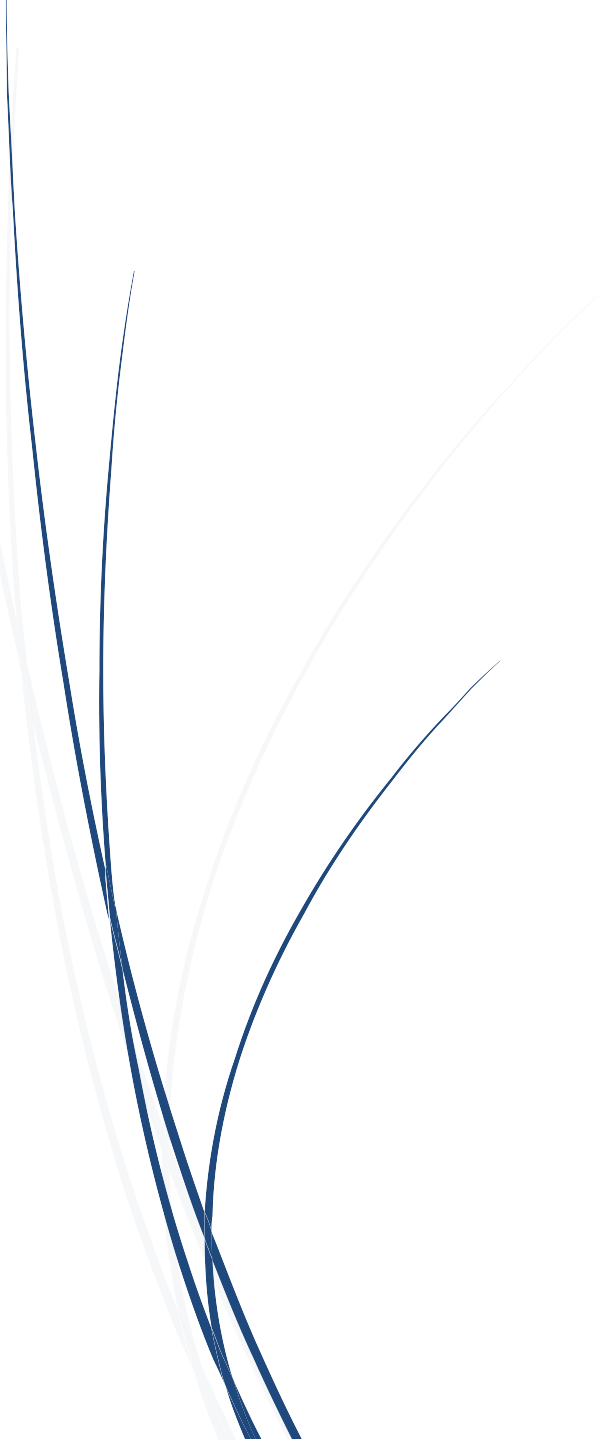
2, 56, 4, -7, 0, 78, -45, 10

Une solution intuitive:

* + Prendre le plus petit
  + Prendre le plus petit suivant
  + …

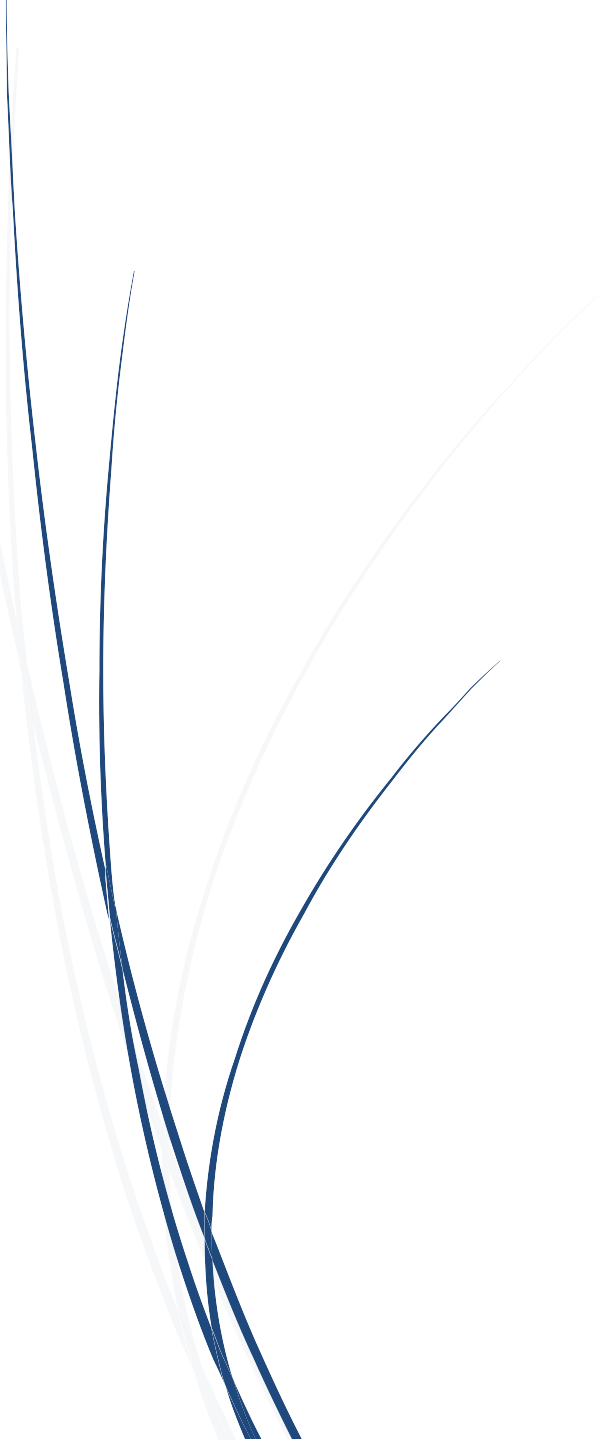
**-45, -7, 0, 2, 4, 10, 56, 78**

# Quelques algorithmes de tri



* + - Tri par Sélection
    - Tri par Insertion
    - Tri à Bulles
    - Tri Rapide
    - Tri Fusion
    - Tri par dénombrement
    - …

Tri par sélection



Déterminer le plus petit élément p à partir de i

Échanger l’élément p avec i

Continuer à partir de i+1

2, 56, 4, -7, 0, 78, -45, 10

-45, 56, 4, -7, 0, 78, 2, 10

-45, 56, 4, -7, 0, 78, 2, 10

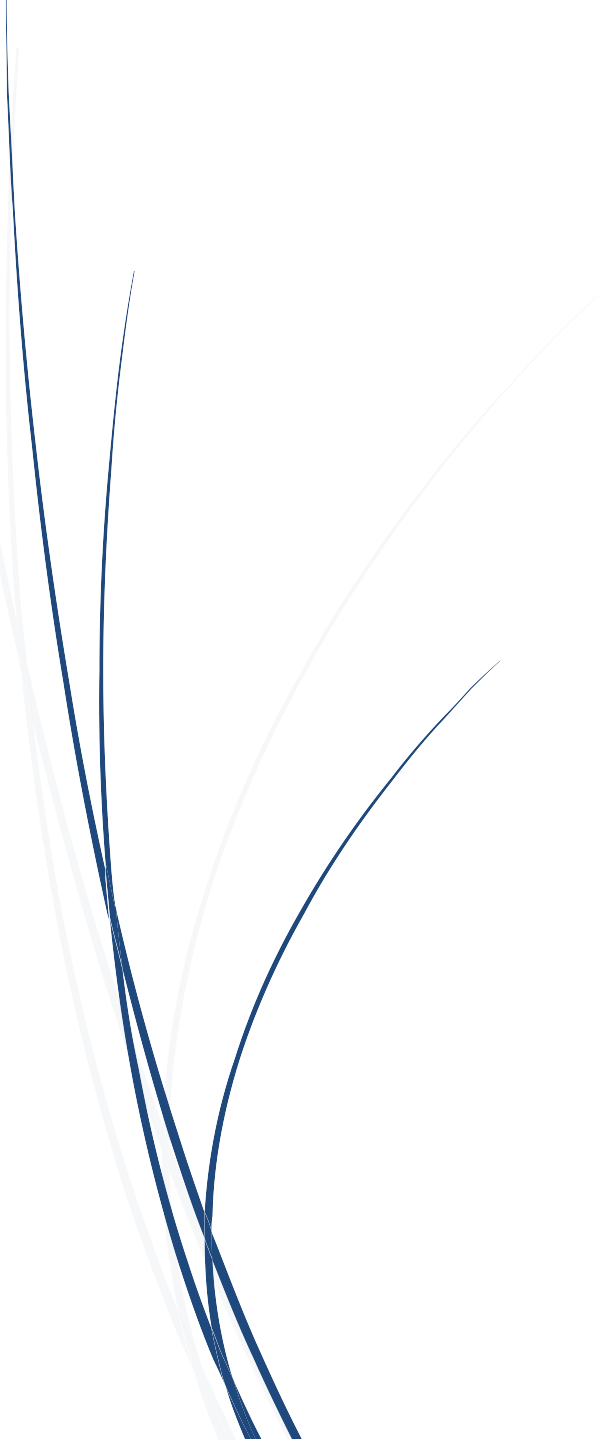
-45, -7, 4, 56, 0, 78, 2, 10

-45, -7, 4, 56, 0, 78, 2, 10

-45, -7, 0, 4, 56, 78, 2, 10

…

# Tri par sélection

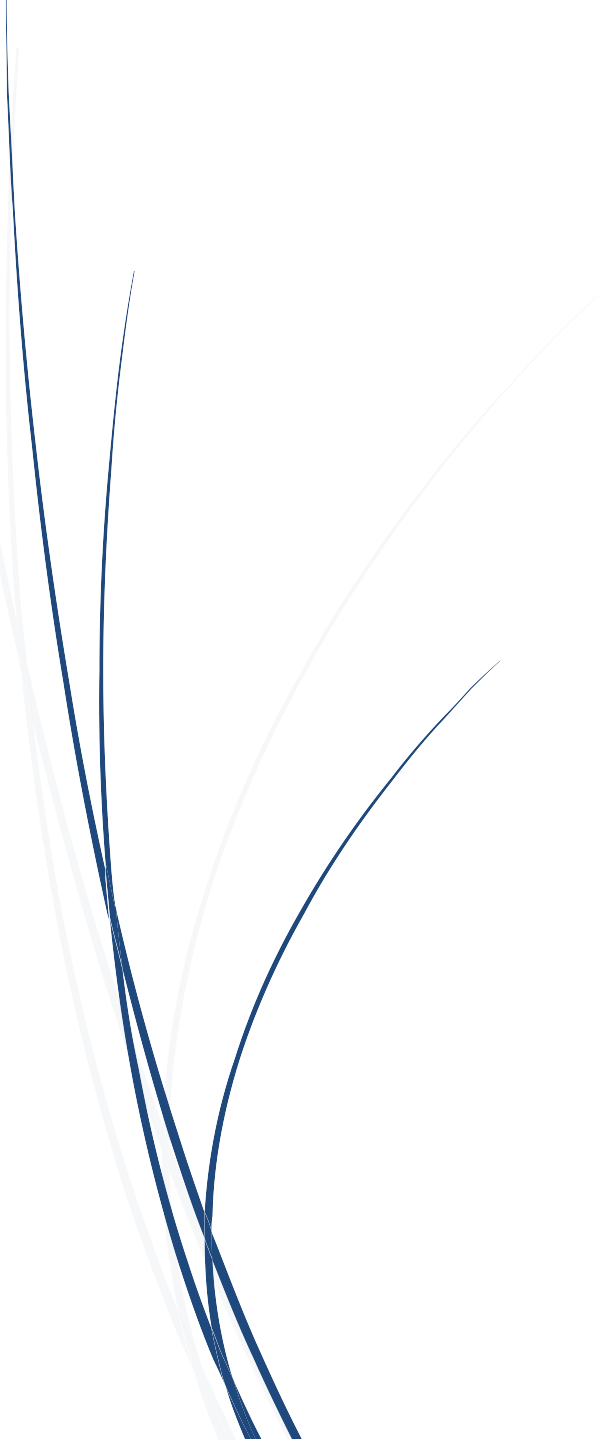


Invariant:

<< A la fin de l’itération i, k contient l’indice de la plus petite valeur de T[i..n] et on échange T[i] et T[k]>>

T[1..i] est trié et il reste à trier T[i+1..n]

Tri par sélection



Algorithme

pour i :=1 à n-1 faire

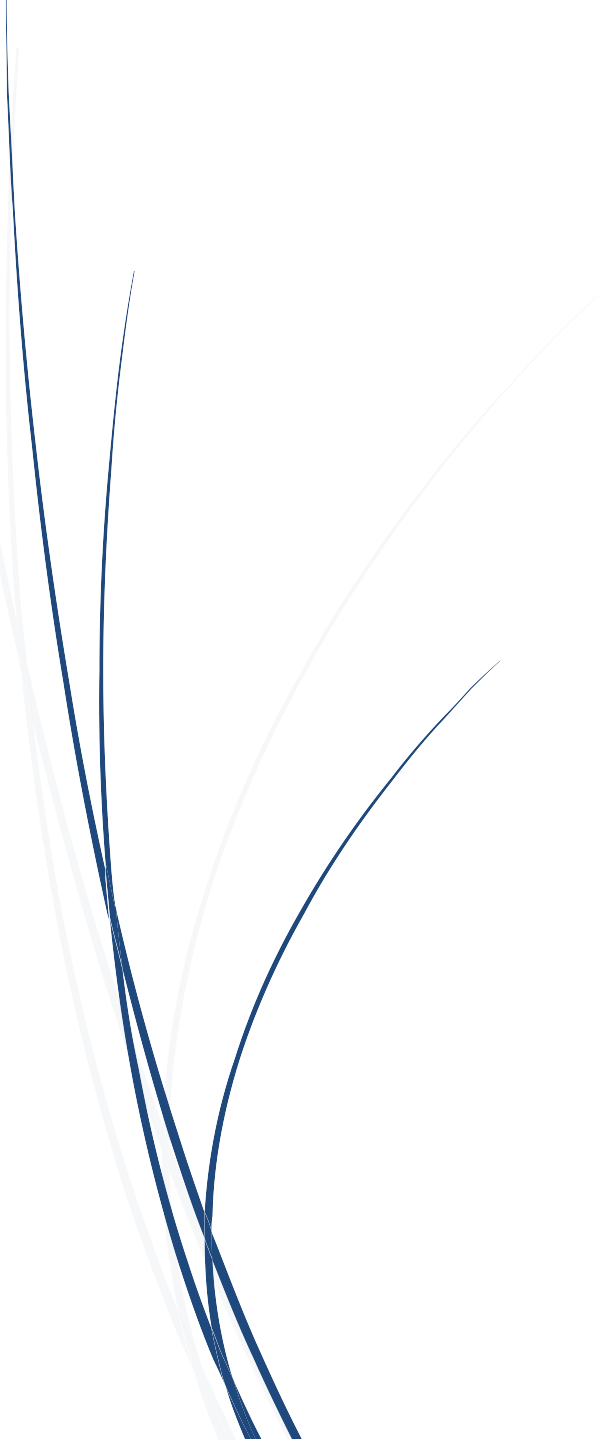
k=minimum(T, i, n) ;

si (i ≠ k) alors échanger (T[i], T[k])

fsi;

fait ;

Tri par insertion



Prendre l’élément i

Insérer i dans l’ordre entre 1 et i

Continuer à partir de i+1

2, 56, 4, -7, 0, 78, -45, 10

2, 56, 4, -7, 0, 78, -45, 10

2, 56, 4, -7, 0, 78, -45, 10

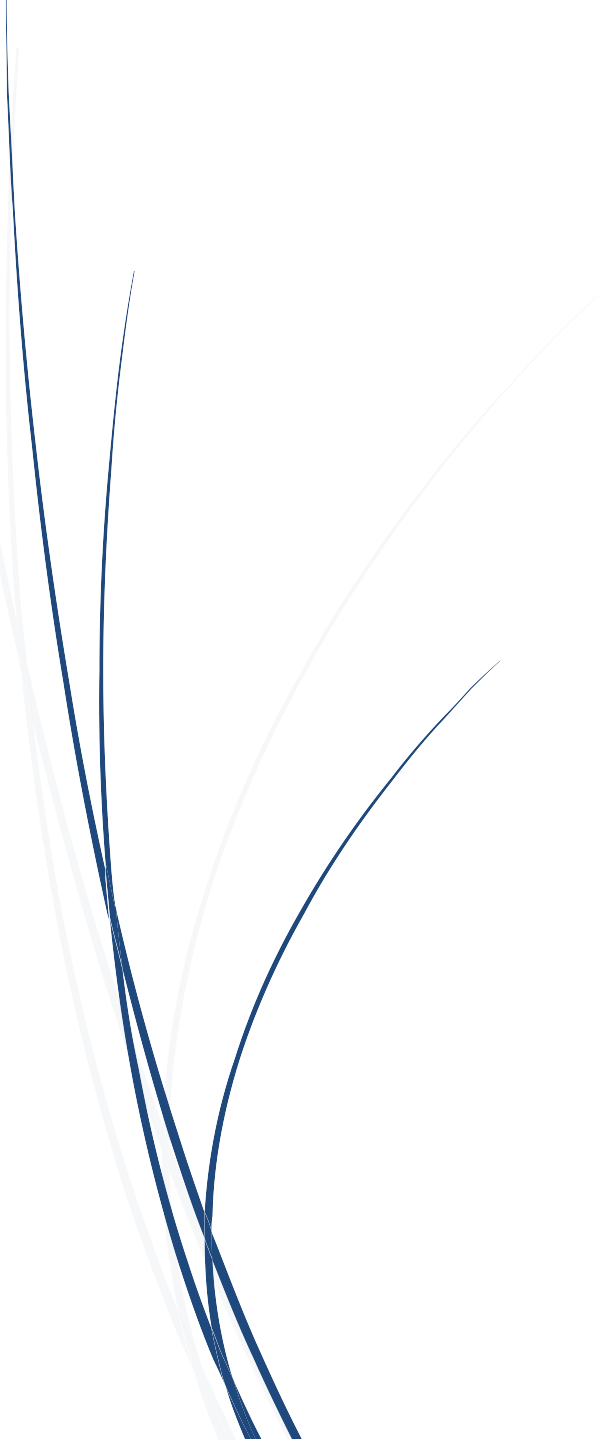
2, 4, 56, -7, 0, 78, -45, 10

-7, 2, 4, 56, 0, 78, -45, 10

-7, 0, 2, 4, 56, 78, -45, 10

…

# Tri par insertion



Algorithme

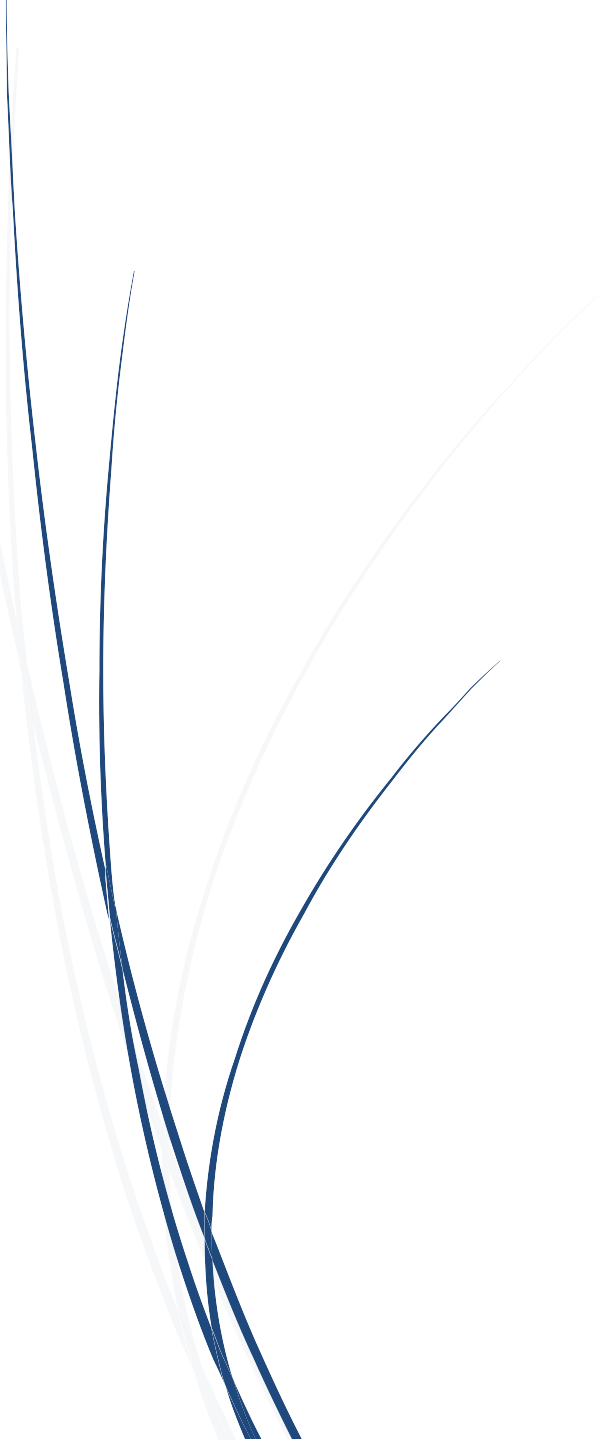
Pour i1 à n faire ji;

tantque (j>1) et (T[j]<T[j-1]) faire échanger(T[j], T[j-1]);

jj-1;

fait; fait;

# Tri par insertion



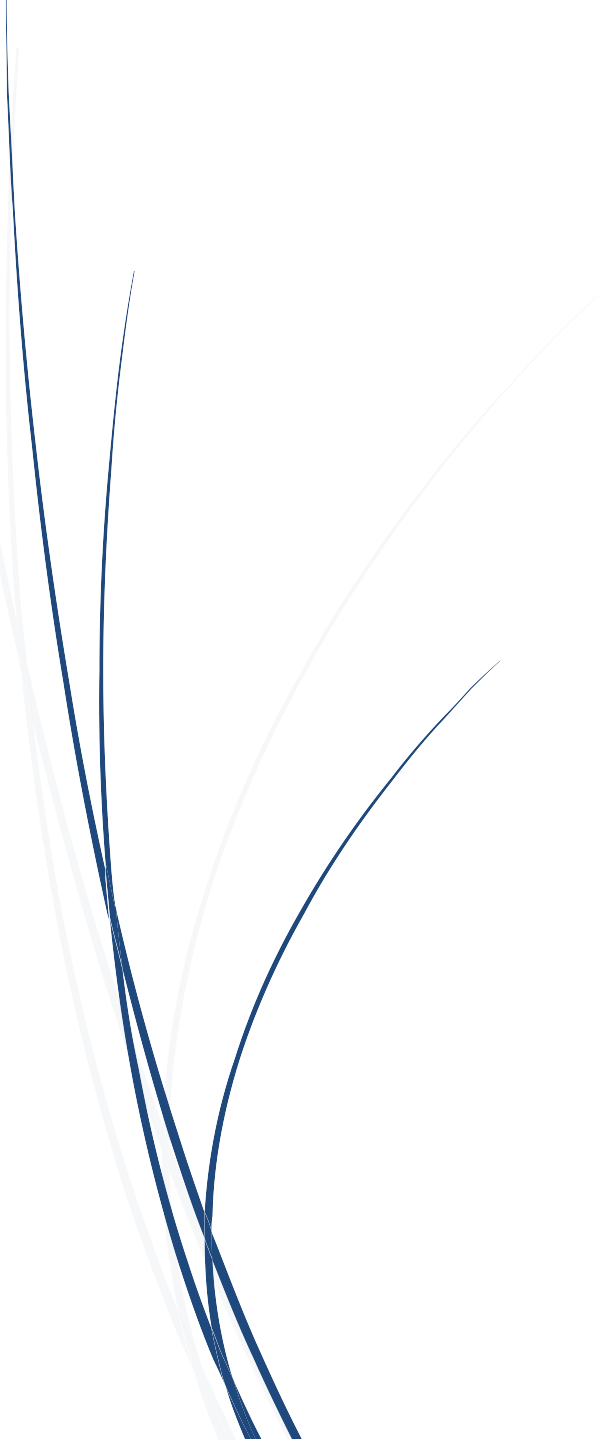
### Invariant de boucle pour:

« A la fin de l’itération i, le sous-tableau T[1..i] est ordonné »

### Invariant de la boucle tantque:

« La boucle tq déplace A[i-1], A[i-2], A[i-3] … d’une position vers la droite jusqu’à trouver la bonne position de A[i] »

# Tri à Bulles



* Comparer 2 à 2 les éléments adjacents
* Les échanger s'ils ne sont pas ordonnés

Comme les bulles, les plus grands éléments remontent en fin de liste.

2, 56, 4, -7, 0, 78, -45, 10

2, 4, -7, 0, 56, -45, 10, 78

2, -7, 0, 4, -45, 10, 56, 78

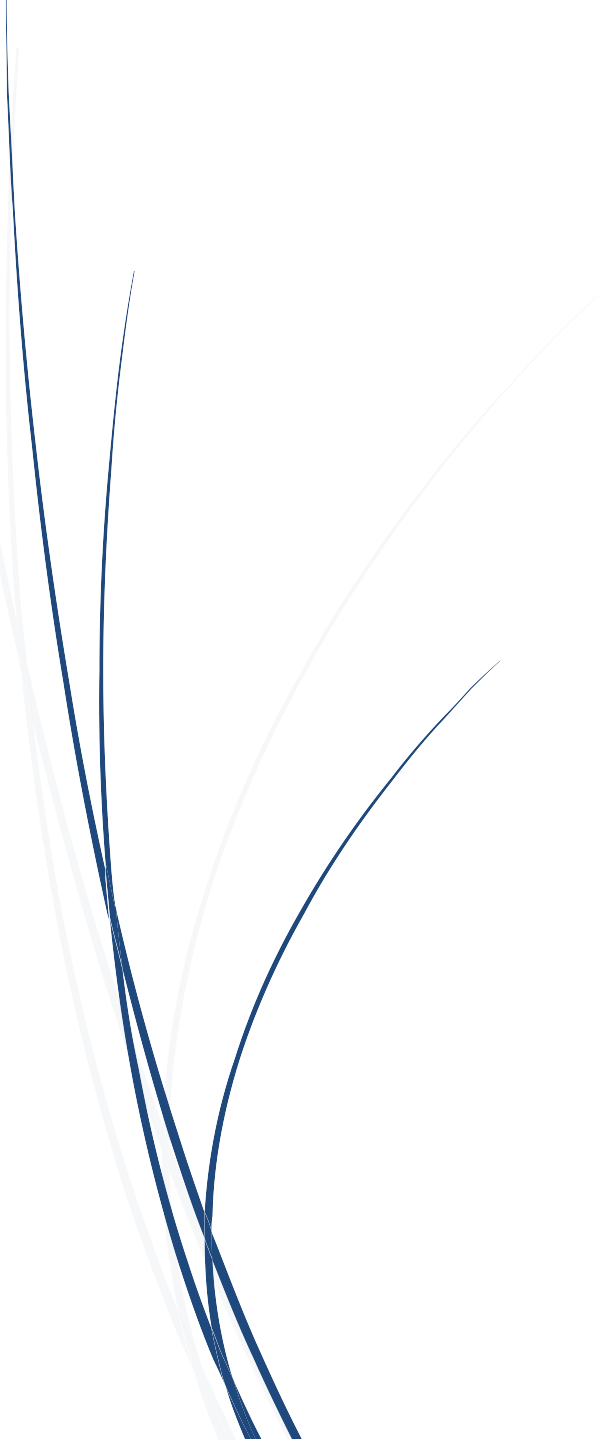
-7, 0, 2, -45, 4, 10, 56, 78

-7, 0, -45, 2, 4, 10, 56, 78

-7, -45, 0, 2, 4, 10, 56, 78

-45, -7, 0, 2, 4, 10, 56, 78

# Tri à Bulles



Algorithme

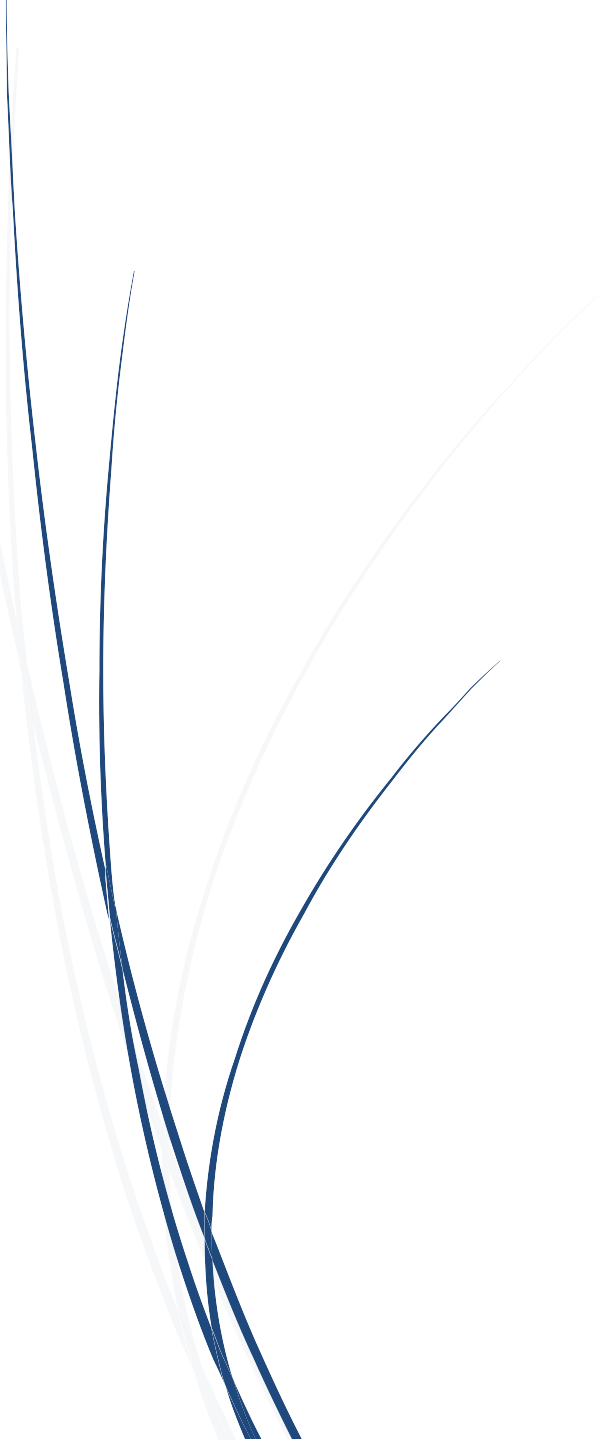
Pour i :=1 à n-1 faire Pour j :=1 à n-i faire

Si (A[j]> A[j+1]) alors échanger(A[j], A[j+1]);

Fsi ; Fait ;

Fait ;

# Tri à Bulles optimisé



boolvrai; i1;

/\* quand il n’y a aucun échange le tableau est trié\*/ Tantque (bool=vrai) faire

boolfaux;

Pour j :=1 à n-i faire

Si (A[j]> A[j+1]) alors échanger(A[j], A[j+1]); boolvrai;

Fsi ; Fait ;

ii+1;

Fait ;