

## I. Définition d'un tri

- Pour trier un tableau, ses valeurs doivent pouvoir être ordonnées
- Un tri doit échanger des valeurs d'un tableau

**procédure** trier (E/S  $t$  : *Tableau*[1..MAX] d'*Element*, E *nbElements* : *Naturel*)

## II. Tris courants

### 1. Tri à bulle

On parcourt le tableau tant qu'il n'est pas trié en inversant les deux premiers éléments consécutifs non ordonné jusqu'à ce que le tableau soit trié.

### 2. Tri par sélection (ou par minimum successif)

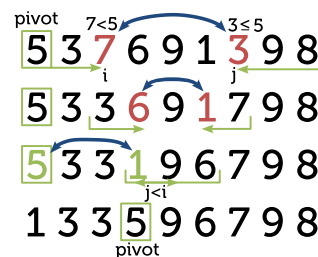
On parcourt le tableau. A chaque indice  $i$ , on parcourt le tableau de droite et on échange le plus petit élément qu'il contient avec la case suivant le tableau gauche.

### 3. Tri par insertion

On parcourt le tableau. A chaque indice  $i$ , on insère l'élément  $i$  où il faut dans le tableau gauche après avoir décalé la zone du tableau entre  $i$  et la nouvelle position de l'élément.

### 4. Tri rapide (récuratif, « intelligent » à la division)

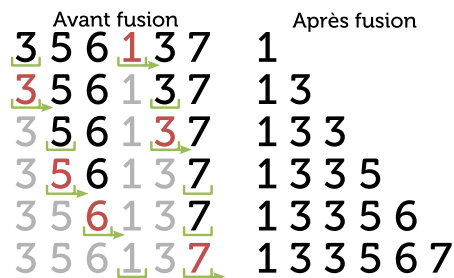
On partitionne le tableau pour que tous les éléments à gauche soient plus petits que tous les éléments à droite d'une valeur pivot. On trie récursivement de la même manière les tableaux de gauche et droite.



Pour partitionner, on prend la première valeur comme pivot, on incrémente  $i$  du début et on décrémente  $j$  de la fin jusqu'à une valeur respectivement plus grande et plus petite ou égale au pivot et on inverse les valeurs des indices  $i$  et  $j$ . On continue jusqu'à ce que  $j$  soit inférieur à  $i$ . Finalement, on inverse la valeur pivot et la valeur de la case  $j$ .

### 5. Tri fusion (récuratif, « intelligent » à la combinaison)

On coupe le tableau en deux, on trie les deux sous-tableaux, puis on les rassemble. Pour trier un sous-tableau, on répète l'opération. Pour fusionner deux tableaux triés, on prend à chaque fois la plus petite valeur entre la première valeur (pas encore prise) du premier tableau et celle du deuxième, jusqu'à avoir tout pris.



## III. Pseudo-code des tris courants

### 1. Tri à bulle $O(n^2)$ $\theta(n^2)$ $\Omega(n)$

**procédure** triABulles (E/S  $t$  : *Tbl*, E *nb* : *Naturel*)

**Declaration**  $i$  : *Naturel*  
 $estTrie$  : *Booleen*

```

début
  répéter
    estTrie ← vrai
    pour  $i \leftarrow 1$  à  $nb - 1$  faire
      si  $t[i] > t[i + 1]$  alors
        échanger( $t[i]$ ,  $t[i + 1]$ )
        estTrie ← faux
      finsi
    finpour
  jusqu'à ce que estTrie
fin
    
```

## 2. Tri par sélection $O(n^2)$ $\theta(n^2)$ $\Omega(n^2)$

**fonction** indiceDuMinimum ( $t : Tbl$ , borneInf, borneSup : *Naturel*) : *Naturel*

**Déclaration**  $i$ , resultat : *Naturel*  
**début**  
 resultat  $\leftarrow$  borneInf  
**pour**  $i \leftarrow$  borneInf + 1 à borneSup **faire**  
     **si**  $t[i] < t[\text{resultat}]$  **alors**  
         resultat  $\leftarrow$   $i$   
     **fin**  
**finpour**  
**retourner** resultat  
**fin**

**procédure** triParSelection (E/S  $t : Tbl$ , E nb : *Naturel*)  
**Déclaration**  $i$  : *Naturel*

**début**  
**pour**  $i \leftarrow$  1 à nb - 1 **faire**  
     echanger( $t[i]$ ,  $t[\text{indiceDuMinimum}(t, i, nb)]$ )  
**finpour**  
**fin**

## 4. Tri fusion $O(n \log_2 n)$ $\theta(n \log_2 n)$ $\Omega(n \log_2 n)$

**procédure** triFusion (E/S  $t : Tbl$ , E nb : *Naturel*)

**début**  
 triFusionRecuratif( $t$ , 1, nb)  
**fin**

**procédure** triFusionRecuratif (E/S  $t : Tbl$ , E d, f : *Naturel*) **fin**

**début**  
**si**  $d < f$  **alors**  
     triFusionRecuratif( $t$ , d,  $(d + f) \text{ div } 2$ )  
     triFusionRecuratif( $t$ ,  $((d + f) \text{ div } 2) + 1$ , f)  
     fusionner( $t$ , d,  $(d + f) \text{ div } 2 + 1$ , f)  
**fin**  
**fin**

**procédure** fusionner (E/S  $t : Tbl$ ; E debut, milieu, fin : *Naturel*)

**Déclaration**  $i$ ,  $j$ ,  $k$  : *Naturel*, temp : *Tbl*  
**début**  
 $i \leftarrow$  debut  
 $j \leftarrow$  milieu  
**pour**  $k \leftarrow$  1 à fin - debut + 1 **faire**  
     **si**  $i \leq$  milieu - 1 et  $j \leq$  fin **alors**  
         **si**  $t[i] \leq t[j]$  **alors**  
             temp[ $k$ ]  $\leftarrow$   $t[i]$   
              $i \leftarrow$   $i + 1$   
         **sinon**  
             temp[ $k$ ]  $\leftarrow$   $t[j]$   
              $j \leftarrow$   $j + 1$   
         **fin**  
     **sinon**  
         **si**  $i \leq$  milieu - 1 **alors**  
             temp[ $k$ ]  $\leftarrow$   $t[i]$   
              $i \leftarrow$   $i + 1$   
         **sinon**  
             temp[ $k$ ]  $\leftarrow$   $t[j]$   
              $j \leftarrow$   $j + 1$   
         **fin**  
     **fin**  
     temp[ $k$ ]  $\leftarrow$  temp[ $k$ ]  
**finpour**  
**fin**

## 3. Tri par insertion $O(n^2)$ $\theta(n^2)$ $\Omega(n^2)$

**fonction** obtenirIndiceDInsertion ( $t : Tbl$ , borneSup : *Naturel*, lEntier : *Entier*) : *Naturel*

**Déclaration**  $i$  : *Naturel*  
**début**  
 $i \leftarrow$  1  
**tant que**  $i <$  borneSup et  $t[i] \leq$  lEntier **faire**  
      $i \leftarrow$   $i + 1$   
**fin**  
**retourner**  $i$   
**fin**

**procédure** decaler (E/S  $t : Tbl$ , E borneInf, borneSup : *Naturel*)

**Déclaration**  $i$  : *Naturel*  
**début**  
**pour**  $i \leftarrow$  borneSup à borneInf + 1 pas de -1 **faire**  
      $t[i] \leftarrow t[i - 1]$   
**finpour**  
**fin**

**procédure** triParInsertion (E/S  $t : Tbl$ , E nb : *Naturel*)

**Déclaration**  $i$ ,  $j$  : *Naturel*, temp : *Entier*  
**début**  
**pour**  $i \leftarrow$  2 à nb **faire**  
      $j \leftarrow$  obtenirIndiceDInsertion( $t$ ,  $i$ ,  $t[i]$ )  
     temp  $\leftarrow$   $t[i]$   
     decaler( $t$ ,  $j$ ,  $i$ )  
      $t[j] \leftarrow$  temp  
**finpour**  
**fin**

## 5. Tri rapide $O(n^2)$ $\theta(n \log_2 n)$ $\Omega(n \log_2 n)$

**procédure** triRapide (E/S  $t : Tbl$ , E nb : *Naturel*)

**début**  
 triRapideRecuratif( $t$ , 1, nb)  
**fin**

**procédure** triRapideRecuratif (E/S  $t : Tbl$ , E d, f : *Naturel*)

**Déclaration** indicePivot : *Naturel*  
**début**  
**si**  $d < f$  **alors**  
     partitionner( $t$ , d, f, indicePivot)  
     triRapideRecuratif( $t$ , d, indicePivot - 1)  
     triRapideRecuratif( $t$ , indicePivot + 1, f)  
**fin**  
**fin**

**procédure** partitionner (E/S  $t : Tbl$ , E debut, fin : *Naturel*; S indicePivot : *Naturel*)

**Déclaration**  $i$ ,  $j$ , pivot : *Naturel*  
**début**  
 pivot  $\leftarrow$   $t[\text{debut}]$   
 $i \leftarrow$  debut  
 $j \leftarrow$  fin  
**tant que**  $i \leq j$  **faire**  
     **si**  $t[i] \leq$  pivot **alors**  
          $i \leftarrow$   $i + 1$   
     **sinon**  
         **si**  $t[j] >$  pivot **alors**  
              $j \leftarrow$   $j - 1$   
         **sinon**  
             echanger( $t[i]$ ,  $t[j]$ )  
         **fin**  
     **fin**  
     indicePivot  $\leftarrow$   $j$   
     echanger( $t[\text{debut}]$ ,  $t[j]$ )  
**fin**