

Algorithmes de tri

NF01

Philippe TRIGANO



Problème

- **Trier un tableau d'éléments de même type**
Le type des éléments doit être muni d'une relation d'ordre

- **Exemple**

Avant : 8 12 5 35 21 3

Après : 3 5 8 12 21 35

Tri par sélection

➤ Principe

- On suppose qu'on connaît le nb d'éléments n
- On cherche le minimum
- On le place en premier élément (on échange)
- On cherche le minimum suivant
- On le place en 2^{ème} position
- Et ainsi de suite

Simulation

| | | | | | |
|---|----|----|----|----|----|
| 8 | 12 | 5 | 35 | 21 | 3 |
| 3 | 12 | 5 | 35 | 21 | 8 |
| 3 | 5 | 12 | 35 | 21 | 8 |
| 3 | 5 | 8 | 35 | 21 | 12 |
| 3 | 5 | 8 | 12 | 21 | 35 |
| 3 | 5 | 8 | 12 | 21 | 35 |

Algorithme

```
pour i allant de 1 à n-1
  min <- T[i]
  indiceMin <- i
  pour j allant de i+1 à n
    si T[j] < min alors
      min <- T[j]
      indiceMin <- j
    fin si
  fin pour
  Echanger T[i] et T[indiceMin]
fin pour
```



Tri par insertion

➤ Principe

- On considère qu'une partie des éléments est ordonnée
- On insère au bon endroit

Algorithme

On cherche le min, on le met au début.

Pour i allant de 2 à n-1

elt \leftarrow T[i+1]

k \leftarrow i

tant que elt < T[k] faire

T[k+1] \leftarrow T[k]

k \leftarrow k-1

fin tant que

T[k+1] \leftarrow elt

fin pour



Simulation

| | | | | | |
|---|----|----|----|----|----|
| 8 | 12 | 5 | 35 | 21 | 3 |
| 3 | 12 | 5 | 35 | 21 | 8 |
| 3 | 5 | 12 | 35 | 21 | 8 |
| 3 | 5 | 12 | 35 | 21 | 8 |
| 3 | 5 | 12 | 21 | 35 | 8 |
| 3 | 5 | 8 | 12 | 21 | 35 |

Tri par échange (tri à bulles)

➤ Principe

- On échange les éléments 2 à 2 en les réordonnant

=> Les éléments mal classés remontent dans la liste comme des bulles à la surface d'un liquide

➤ Efficacité

- Dépend du tableau initial
- Efficace si le tableau est presque trié

Algorithme

```
nbElts ← n      {nb elts restant à classer }  
echange ← vrai  {echange est vrai s'il y a eu un echange}  
tant que echange faire  
    echange ← faux  
    max ← nbelts  
    pour i allant de 1 à max-1 faire  
        si T[i] > T[i+1] alors  
            echanger T[i] et T[i+1]  
            echange ← vrai  
            nbelts ← i  
        fsi  
    fpour  
ftq
```

