

Algorithmes de tri

EA3

2015--2016

F. Laroussinie

Tri par sélection

```
# IndiceMin(T,i) retourne l'indice de l'élément min de T[i]...T[|T|-1]
# hyp:  $0 \leq i < \text{len}(T)$ 
```

```
def IndiceMin(T,i) :
    res = i
    Pour k = i+1... |T|-1 :
        Si  $T[k] < T[\text{res}]$  Alors res = k
    Retourner res
```

```
def TriSelection(T) :
    Pour i = 0.. |T|-2 :
        | ind = IndiceMin(T,i)
        |  $T[i] \leftrightarrow T[\text{ind}]$ 
    Retourner T
```

Tri par insertion

```
# inserer T[i] à sa place dans T[0...i-1]
```

```
# i = 1...n-1
```

```
def Insérer(T,i) :
```

```
    Si  $i > 0$  Alors :
```

```
        Si  $T[i-1] > T[i]$  Alors :
```

```
             $T[i-1] \leftrightarrow T[i]$ 
```

```
            Insérer(T,i-1)
```

```
def TriInsertion(T) :
```

```
    Pour  $i = 1 \dots |T|-1$  :
```

```
        Insérer(T,i)
```

```
    Retourner T
```

*version avec plusieurs
permutations pour placer T[i]*

Tri par insertion

```
# inserer T[i] à sa place dans T[0...i-1]
```

```
# i = 1...n-1
```

```
def Insérer(T,i) :
```

```
  val = T[i]
```

```
  j=i
```

```
  tant que j>0 et T[j-1]>val :
```

```
    T[j]=T[j-1]
```

```
    j=j-1
```

```
  T[j]=val
```

```
def TriInsertion(T) :
```

```
  Pour i = 1...|T|-1 :
```

```
    Insérer(T,i)
```

```
  Retourner T
```

*ici on attend de trouver le
bon endroit où placer val (=T[i])
pour l'y placer...*