

Tri par insertion

idée: Garder le début du tableau trié. Y insérer successivement les éléments restants.

algo:

TriInsertion(T):

$n \leftarrow \text{longueur}(T)$

pour  $i$  de 1 à  $n-1$ :

    TriPartiel(T, i)

TriPartiel(T, i)

$g \leftarrow i$

while ( $g > 0$  &&  $T[g] < T[g-1]$ )

    échanger  $T[g]$  et  $T[g-1]$

$g--$

ex:

$i=1$      $1 < 10$

$i=2$      $5 < 10$   
           $5 > 1$

$i=3$      $19 > 10$

$i=4$      $3 < 19$   
           $3 < 10$   
           $3 < 5$   
           $3 > 1$

$i=5$      $3 < 19$   
           $3 < 10$   
           $3 < 5$   
           $3 = 3$

$\{10, 1, 5, 19, 3, 3\}$

$\{1, 10, 5, 19, 3, 3\}$

$\{1, 5, 10, 19, 3, 3\}$   
 $\{1, 5, 10, 19, 3, 3\}$

$\{1, 5, 10, 19, 3, 3\}$

$\{1, 5, 10, 3, 19, 3\}$   
 $\{1, 5, 3, 10, 19, 3\}$   
 $\{1, 3, 5, 10, 19, 3\}$   
 $\{1, 3, 5, 10, 19, 3\}$

$\{1, 3, 5, 10, 3, 19\}$   
 $\{1, 3, 5, 3, 10, 19\}$   
 $\{1, 3, 3, 5, 10, 19\}$   
 $\{1, 3, 3, 5, 10, 19\}$

Un tri est **STABLE**  
s'il **préserve** l'ordre  
des éléments qui  
sont considérés **égaux**

Complexité

meilleur des cas : linéaire  $\rightarrow \sim n$

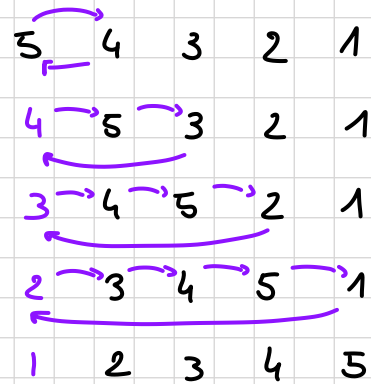
pire des cas : quadratique  $\rightarrow \frac{(n-1)n}{2} \sim n^2$

## TD3

### Exercice 1

8	4	10	-5	1
4	8	10	-5	1
4	8	-5	10	1
4	-5	8	10	1
-5	4	8	10	1
-5	4	8	1	10
-5	4	1	8	10
-5	1	4	8	10

4 comparaisons  
pas d'affectations



13 comparaisons  
39 affectations.

7 comparaisons

21 affectations

(il faut une variable tmp)

### Exercice 2

while ( $g > 0 \ \&\& \text{Inf}(T[g], T[g-1])$ )

Inf(a, b)

if ( $a \% 2 == 1$ ) {

if ( $b \% 2 == 1$ ) return  $a < b$

else return False // b est pair donc reste à gauche

} else {

if ( $b \% 2 == 1$ ) return True // a est pair donc doit aller à gauche de b

else return  $a < b$

}

### Exercice 3

Inf(a, b):

if (a.couleur < b.couleur) return True

if (a.couleur > b.couleur) return False

else return a.rang < b.rang

### Exercice 4

1) a.note = 10      alors  $a \leq b$  et  $b \leq a$   
b.note = 10

2) insertion: oui /

sélection: non /

### Exercice 5

1)  $\boxed{n-1=2}$       3   1   2   0

i = 0 < 2    3 > 1    1   3   2   0

i = 0 < 2    1 < 3

i = 1 < 2    3 > 2    1   2   3   0

i = 0 < 2    1 < 3

i = 1 < 2    2 < 3

i = 2     $\boxed{\text{STOP}}$

2) L'algo trie les (n-1) premiers éléments du tableau

3) 5 comparaisons dans la boucle while.

## Exercice 6

1)  $Tri(T)$ :

// trouver la frontière

int  $f = 0$

while ( $T[f] < T[f+1]$ ) {  $f++$ ; }

// Tri Insertion à partir de  $f$ .

pour  $i = f+1$  à  $n-1$  {

$g \leftarrow i$

while ( $g > 0$  &&  $T[g] < T[g-1]$ ) {

échanger  $T[g]$  et  $T[g-1]$

$g--$

} }

2)  $T$  = fusion des tableaux  $A$  et  $B$  de tailles respectives  $m$  et  $l$

Dans le pire des cas :  $m \cdot l$  comparaisons.

↳ tous les éléments de  $B$  sont plus petits que le  $\oplus$  gd de  $A$