

# TD 14

## Tri à bulles (Bubble sort)

Algorithmique  
Semestre 1

### 1

#### 1.1 Algo niveau 1

```
1  -- Algo niveau 1
2  se placer sur le premier element a traiter
3  tantque tous les elements n ont pas ete traites faire
4      comparer et traiter deux elements adjacents;
5      passer a l element suivant;
6  fin tantque;
```

Listing 1 – sous-programme ramenerValeurMax Niveau 1

#### 1.2 Algo niveau 2

```
1  -- Algo niveau 2
2
3  se placer sur le premier element a traiter
4  tantque tous les elements n ont pas ete traites faire
5      --comparer et traiter deux elements adjacents
6      si l element courant est superieur a l element suivant alors
7          les permuter;
8      fin si;
9      passer a l element suivant;
10 fin tantque;
```

Listing 2 – sous-programme ramenerValeurMax Niveau 2

#### 1.3 Trace du sous-programme

2	-12	7	3	7	5
---	-----	---	---	---	---

-12	2	7	3	7	5
-----	---	---	---	---	---

-12	2	7	3	7	5
-----	---	---	---	---	---

-12	2	3	7	7	5
-----	---	---	---	---	---

-12	2	3	7	7	5
-----	---	---	---	---	---

-12	2	3	7	5	5
-----	---	---	---	---	---

2	-12	7	3	7	5
---	-----	---	---	---	---

## 2 En-tête de ramenerValeurMax

```

1  -- ramene par permutation successives la plus grande valeur d'un tableau
   tab de ne elements
2  -- en derniere position
3  procedure ramenerValeurMax(maj tab, entree n);

```

Listing 3 – En-tête de ramenerValeurMax

### 2.1 Trace

situation	1	2	3	4	5	6	7	...	n	i	aux	$i \neq n$	$tab[i] > tab[i+1]$
0	2	-12	7	3	5	7	?		6	?	?	?	i
1										1			
2												$1 \neq 6$	
3												?	$2 > -12$
4	-12										2		?
5										2			
2												$2 \neq 6$	
3													$2 \neq 7$
5												$3 \neq 6$	
2													$7 > 3$
3			3	7							7		
4										4			
5												$4 \neq 6$	
2													
3										5			
5												$5 \neq 6$	
2													$7 > 5$
3					5	7					7		
4										6			
5												$6 \neq 6$	
2	-12	2	3	7	5	7							

### 2.2

```

1  -- ramener par permutation succesives la plus grande valeur d'un intervalle
   iDeb iFin d'un tableau d'entier en derniere position (iFin)
2  procedure ramenerValeurMax (maj tab <TabEntier>, entree iDeb <Entier>,
   entree iFin <Entier>);

```

Listing 4 – En-tête ramenerValeurMax

## 3

```

1  ramenerValeurMax(tab, 1, n);

```

Listing 5 – utilisation de ramenerValeurMax

## 4

### 4.1 Algo général

```

1  -- trier un tableau par la methode des bulles
2  tantque il reste un element du tableau a classer faire
3      ramener la plus grande valeur a la fin du tableau;
4      reduire le tableau d'un element;
5  fin tantque;
```

Listing 6 – Algorithme général du tri à bulles

### 4.2 Programme

```

1  procedure trierParBulles (maj tab<TabEntiers>, entree n <Entier>)
2  glossaire
3      nbElements <Entier>;
4  debut
5      nbElements <- n;
6      tantque nbElements > 1 faire
7          ramenerValeurMax (tab,1,nbElements);
8          nbElements <- nbElements - 1;
9      fin tantque;
10 fin
```

Listing 7 – Programme trierParBulles

## 5

Situations	nbElements	1	2	3	4	5	6
1	6	2	-12	7	3	7	5
2	6	-12	2	3	7	7	7
3	5	-12	2	3	7	7	7
2	5	-12	2	3	3	5	7

## 6

### 7 Postcondition

$$\forall i \in [1, n-1], tab[i] < tab[i+1]$$

## 8 Comparaisons

$$\begin{aligned}
 & (n-1) + (1-2) + (1-3) + \dots + 1 \\
 = & \sum_{c=1}^{n-1} (n-1) \\
 = & \sum_{i=n}^{n-1} - \sum_{i=1}^{n-1} i \\
 = & \frac{(n-1)(n+2)}{2} - (n-1) = \frac{n(n-1)}{2} = O(n^2)
 \end{aligned}$$

Donc complexité en  $O(n^2)$