

Tri Bulle

1	2	3	4	5	6
3	2	4	1	5	4

taille : $N = 6$

Tri Bulle

étape 1

1	2	3	4	5	6
3	2	4	1	5	4
2	3	4	1	5	4
2	3	4	1	5	4
2	3	1	4	5	4
2	3	1	4	5	4
2	3	1	4	4	5

Tri Bulle

étape 2

1	2	3	4	5	6
2	3	1	4	4	5
2	3	1	4	4	5
2	1	3	4	4	5
2	1	3	4	4	5
2	3	1	4	4	5

Tri Bulle

étape 3

1	2	3	4	5	6
2	1	3	4	4	5
1	2	3	4	4	5
1	2	3	4	4	5
1	2	3	4	4	5

Tri Bulle

étape 4

1	2	3	4	5	6
1	2	3	4	4	5
1	2	3	4	4	5
1	2	3	4	4	5

Tri Bulle

étape 5

1	2	3	4	5	6
1	2	3	4	4	5
1	2	3	4	4	5

Tri Bulle (version naïve)

R0

(tableau t initialisé avec N valeurs *)*

Pour $i = 1$ **à** $N-1$ **Faire**

traiter le tableau de taille $N - i + 1$ *(* ss pb 1 *)*

(tableau $t(N-i+1 : N)$ trié *)*

Fin Pour

(tableau $t(1:N)$ trié *)*

Tri Bulle (version naïve)

R1

(tableau t initialisé avec N valeurs *)*

Pour $i = 1$ à $N-1$ **Faire**

(traiter le tableau de taille $N - i + 1$ *) (* ss pb 1 *)*

Pour $j = 1$ à $N - i$

traiter les élément $t(j)$ et $t(j+1)$ *(* ss pb 1.1 *)*

($t(j) \leq t(j+1)$ *)*

Fin Pour

(tableau $t(N-i+1 : N)$ trié *)*

Fin Pour

(tableau $t(1:N)$ trié *)*

Tri Bulle (version naïve)

R2

(tableau t initialisé avec N valeurs *)*

Pour $i = 1$ à $N-1$ **Faire**

(traiter le tableau de taille $N - i + 1$ *) (* ss pb 1 *)*

Pour $j = 1$ à $N - i$

(traiter les élément $t(j)$ et $t(j-1)$ *) (* ss pb 1.1 *)*

Si $t(j) > t(j+1)$ **Alors**

échanger $t(j)$ et $t(j+1)$ *(* ss pb 1.1.1 *)*

Fin Si

($t(j) \leq t(j+1)$ *)*

Fin Pour

(tableau $t(N-i+1 : N)$ trié *)*

Fin Pour

(tableau $t(1:N)$ trié *)*

Améliorations

- on peut limiter le nombre d'étapes si on sait détecter que le tableau est déjà trié
 - => un booléen indique que, lors d'une étape, on a effectué au moins une inversion et nous permet de s'arrêter s'il reste à faux
 - => on transforme ainsi la boucle externe en **tant que**
- on peut aussi limiter le nombre de sous-étapes en mémorisant où a eu lieu la dernière inversion lors de l'étape précédente
 - => une variable entière prend le rôle du booléen
 - => on raccourcit la boucle interne