

Chapitre 3 :

Les algorithmes de tri

Pour trier un tableau, on peut utiliser les algorithmes de tri suivants : *Tri par sélection* -- *Tri à bulles* -- *Tri par insertion* -- *Tri de Shell*

1) Le tri par sélection

Principe :

Cette méthode de tri consiste à :

- ⇒ Rechercher la position du minimum (ou maximum si l'ordre est décroissant) de 0 à $n-1$
- ⇒ Comparer ce minimum avec $T[0]$. S'ils sont différents on les permute.
- ⇒ Rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1 (s'ils sont différents)
- ⇒ Continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.

Exemple :

T	20	0	-7	11	25	3
	0	1	2	3	4	5
T	-7	0	20	11	25	3
	0	1	2	3	4	5
T	-7	0	20	11	25	3
	0	1	2	3	4	5
T	-7	0	3	11	25	20
	0	1	2	3	4	5
T	-7	0	3	11	25	20
	0	1	2	3	4	5
T	-7	0	3	11	20	25
	0	1	2	3	4	5

$T[0] \neq T[2]$. On permute $T[0]$ et $T[2]$.

Pas de permutation

$T[2] \neq T[5]$. On permute $T[2]$ et $T[5]$.

Pas de permutation

$T[4] \neq T[5]$. On permute $T[4]$ et $T[5]$.

Tableau Trié.

Procédure tri:selection (n:entier, @T:tab)

Debut

Pour i de 0 à n-2 faire

$p_{min} \leftarrow \text{Position min}(T, n, i)$

si $T[p_{min}] \neq T[i]$ alors

$x \leftarrow T[i]$

$T[i] \leftarrow T[p_{min}]$

$T[p_{min}] \leftarrow x$

fin si

fin pour

fin

T p o l

$\begin{matrix} p_{min} \\ i \\ x \end{matrix} \left. \vphantom{\begin{matrix} p_{min} \\ i \\ x \end{matrix}} \right\} \text{Entier}$

$\begin{matrix} \text{Position min} \\ \text{fonction} \end{matrix}$

fonction position_min(T:tab, n, i:entier):Entier

0 Debut

$P \leftarrow i$

Pour j de i+1 à n-1 faire

si $T[j] < T[i]$ alors

$P \leftarrow j$

fin

fin pour

retourner P

fin

T:tab
0 T[n]
j } entier
P

2) Le tri à bulles

Principe :

Cette méthode de tri consiste à :

- ⇒ Comparer deux à deux les éléments consécutifs du tableau et les permuter s'ils ne sont pas triés.
- ⇒ Après un seul passage sur le tableau, la plus grande valeur est placée dans la dernière case (dans le cas d'un tri croissant), mais le tableau n'est pas entièrement trié.
- ⇒ Refaire l'étape 1 sur la partie non triée du tableau $T[0] \dots T[n-2]$ et ainsi de suite...
- ⇒ On s'arrête : si aucune permutation ne sera faite au cours d'une passe ou si la partie non triée est réduite à un seul élément.

Exemples :

Exemple 1 :

10	6	8	5	4
----	---	---	---	---

Tableau initial : Partie non triée

Echange=Faux

Parcours 1

6	↔ 10	8	5	4
6	8 ↔ 10	5	4	
6	8	5 ↔ 10	4	
6	8	5	4 ↔ 10	
6	8	5	4	10

Partie non triée Partie triée

10>6 permutation Echange=Vrai
10>8 permutation Echange=Vrai
10>5 permutation Echange=Vrai
10>4 permutation Echange=Vrai

Echange=Faux

Parcours 2

6	8	5	4	10
6	5 <=> 8	4	10	10
6	5	4 <=> 8	10	10
6	5	4	8	10

Partie non triée Partie triée

6>8
8>5 permutation Echange=Vrai
8>4 permutation Echange=Vrai

Echange=Faux

Parcours 3

5	6	4
5	4	6
5	4	6

8 10
8 10
8 10

6>5 permutation
6>4 permutation

Echange=Vrai
Echange=Vrai

Partie non triée Partie triée

Echange=Faux

Parcours 4

4	5	6
4	5	6

8 10
8 10

5>4 permutation

Echange=Vrai

Partie non triée Partie triée

La partie non triée se compose d'un seul élément, on arrête le traitement.

Le tableau trié est : 4 | 5 | 6 | 8 | 10

Exemple 2 :

12	5	10	15	19
----	---	----	----	----

Tableau initial : Partie non triée

Echange=Faux

Parcours 1

5	12	10	15	19
5	10	12	15	19
5	10	12	15	19
5	10	12	15	19
5	10	12	15	19

12>5 permutation Echange=Vrai

12>10 permutation Echange=Vrai

12>15

15>19

Partie non triée Partie triée

Echange=Faux

Parcours 2

5	10	12	15	19
5	10	12	15	19
5	10	12	15	19
5	10	12	15	19

5>10

10>12

12>15

Partie non triée Partie triée

Parcours 2 terminé sans aucune permutation (la variable Echange reste Faux) donc le tableau est complètement trié.

On arrête le traitement. Le tableau trié est : 5 | 10 | 12 | 15 | 19

4 SI 2 -- Algo&Prog

Enseignante : ICHRAF MABROUK

Procédure tri-bulle (n-entier, T-tab)

début

Repete

echange ← faux

Parcours de 0 à n-1 faire

si $T[i] > T[i+1]$ alors

$x \leftarrow T[i]$

$T[i] \leftarrow T[i+1]$

$T[i+1] \leftarrow x$

echange ← Vrai

fin pour

$n \leftarrow n-1$

jusqu'à echange = faux ou $n=1$

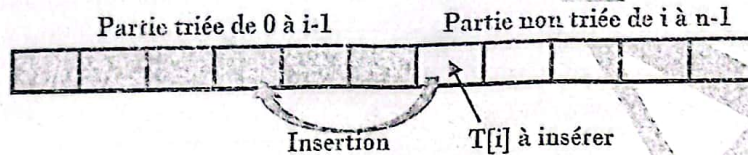
3) Le tri par insertion

Principe :

On suppose que le tableau est partagé en deux parties : partie triée et partie non triée. Initialement la partie triée contient le premier élément.

Pour chaque élément d'indice i on cherche à l'insérer dans sa bonne position dans la partie triée. Pour cela :

- ⇒ Stocker sa valeur dans une variable intermédiaire x .
- ⇒ Décaler tant que l'élément x est inférieur à son précédent.
- ⇒ Affecter à la dernière case décalée la valeur de la variable intermédiaire x .



Exemple :

T	17	9	0	-7	11	9
	0	1	2	3	4	5

Etape1 : x contient $T[1] \Rightarrow x = 9$

Etape2 : Décalage :

- 1) $T[0] > x$ donc $T[1]$ contient $T[0]$

Etape3 : $T[0]$ contient le contenu de la variable x

x 9

T	9	17	0	-7	11	9
	0	1	2	3	4	5

Etape1 : x contient $T[2] \Rightarrow x = 0$

Etape2 : Décalage :

- 1) $T[1] > x$ donc : $T[2]$ contient $T[1]$
- 2) $T[0] > x$ donc : $T[1]$ contient $T[0]$

Etape3 : $T[0]$ contient le contenu de la variable x

x 0

T	0	9	17	-7	11	9
	0	1	2	3	4	5

Etape1 : x contient $T[3] \Rightarrow x = -7$

Etape2 : Décalage :

- 1) $T[2] > x$ donc : $T[3]$ contient $T[2]$
- 2) $T[1] > x$ donc : $T[2]$ contient $T[1]$
- 3) $T[0] > x$ donc : $T[1]$ contient $T[0]$

Etape3 : $T[0]$ contient le contenu de la variable x

x -7

T	-7	0	9	17	11	9
	0	1	2	3	4	5

Etape1 : x contient $T[4] \Rightarrow x = 11$

Etape2 : Décalage :

- 1) $T[3] > x$ donc : $T[4]$ contient $T[3]$

Etape3 : $T[3]$ contient le contenu de la variable x

x 11

T	-7	0	9	11	17	9
	0	1	2	3	4	5

Etape1 : x contient $T[5] \Rightarrow x = 9$

Etape2 : Décalage :

- 1) $T[4] > x$ donc : $T[5]$ contient $T[4]$
- 2) $T[3] > x$ donc : $T[4]$ contient $T[3]$

Etape3 : $T[3]$ contient le contenu de la variable x .

x 9

T	-7	0	9	9	11	17
	0	1	2	3	4	5

Procédure tri insertion (n : entier, T : tab)

Debut

Pour i de 1 à $n-1$ faire

$x \leftarrow T[i]$

$j \leftarrow i$

Tant que $j > 0$ et $T[j-1] > x$ faire

$T[j] \leftarrow T[j-1]$

$j \leftarrow j-1$

fin tant que

$T[j] \leftarrow x$

fin pour

T D O K

o	T
i	entier
x	entier
j	entier