

I21: Introduction à l'algorithmique

Cours 4: Algorithmes de tris

Nicolas Méloni

Licence 1: 2ème semestre
(2017/2018)

Problème : Tri

Entrée : tableau d'entiers T de taille n

Sortie : une permutation des éléments de T telle que

$$T[1] \leq T[2] \leq \dots \leq T[n].$$

- ❑ Trier des données est la base de nombreux algorithmes :
 - ❑ recherche d'élément
 - ❑ plus proche paire
 - ❑ unicité
 - ❑ enveloppe convexe
- ❑ Les algorithmes de tris font appels à des idées réutilisables dans de nombreux autres contextes.
- ❑ C'est historiquement le problème le plus étudié en informatique.

Trois algorithmes *classiques* :

- tri par sélection
- tri par propagation
- tri par insertion

Ces tris partagent 2 grandes caractéristiques :

Tris Comparatifs

Ils reposent exclusivement sur l'opération de comparaison des éléments.

Tris in situ

Seul un nombre constant d'éléments est stocké hors du tableau à trié.

L'opération principale est l'échange de deux valeurs du tableau.

```
1  ALGORITHME Swap(T, i, j):  
2  DONNEES  
3    T: tableau d entiers  
4    i, j: entiers  
5  VARIABLES:  
6    aux: entier  
7  DEBUT  
8    aux ← T[i]  
9    T[i] ← T[j]  
10   T[j] ← aux  
11 FIN
```

■ complexité : $O(1)$

Idée générale :

- ❏ Chercher le plus petit élément du tableau ;
- ❏ l'échanger avec l'élément d'indice 1 ;
- ❏ chercher le deuxième plus petit élément ;
- ❏ l'échanger avec l'élément d'indice 2 ;
- ❏ etc

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---



Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---



1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---

Diagram illustrating the first step of selection sort. The array is [1, 6, 8, 4, 7, 5, 2, 3]. The element 1 is the current minimum (highlighted in light blue), and 4 is the element being compared to it (highlighted in green). A curved arrow points from 4 to 1, indicating that 4 is swapped with 1.

1	2	8	4	7	5	6	3
---	---	---	---	---	---	---	---

Diagram illustrating the second step of selection sort. The array is [1, 2, 8, 4, 7, 5, 6, 3]. The element 1 is the current minimum (highlighted in light blue), and 6 is the element being compared to it (highlighted in green). A curved arrow points from 6 to 1, indicating that 6 is swapped with 1.

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---



1	2	8	4	7	5	6	3
---	---	---	---	---	---	---	---



1	2	8	4	7	5	6	3
---	---	---	---	---	---	---	---

Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

Tri par sélection

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

1	6	8	4	7	5	2	3
---	---	---	---	---	---	---	---

Arrow from index 0 to index 3

1	2	8	4	7	5	6	3
---	---	---	---	---	---	---	---

Arrow from index 1 to index 6

1	2	3	4	7	5	6	8
---	---	---	---	---	---	---	---

Arrow from index 2 to index 7

1	2	3	4	7	5	6	8
---	---	---	---	---	---	---	---

Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

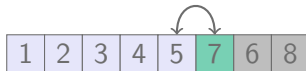
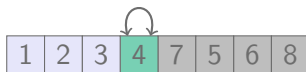
1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

1 2 3 4 5 7 6 8

Tri par sélection



Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

1 2 3 4 5 7 6 8

1 2 3 4 5 6 7 8

Tri par sélection

4 6 8 1 7 5 2 3

1 6 8 4 7 5 2 3

1 2 8 4 7 5 6 3

1 2 3 4 7 5 6 8

1 2 3 4 7 5 6 8

1 2 3 4 5 7 6 8

1 2 3 4 5 6 7 8

1 2 3 4 5 6 7 8

Tri par sélection

```
1  ALGORITHME TriSelection(T):  
2  DONNEES  
3    T: tableau d'entiers de taille n  
4  VARIABLES  
5    imin, i, j: entiers  
6  DEBUT  
7    i ← 1  
8    TQ i ≤ n-1 FAIRE  
9      imin ← i  
10     j ← i+1  
11     TQ j ≤ n FAIRE  
12       SI T[j] < T[imin] ALORS  
13         imin ← j  
14       FSI  
15     j ← j+1  
16     FTQ  
17     Swap(T, i, imin)  
18     i ← i+1  
19   FTQ  
20  FIN
```

Tri par sélection

```
1  ALGORITHME TriSelection(T):
2  DONNEES
3  T: tableau d'entiers de taille n
4  VARIABLES
5  imin, i, j: entiers
6  DEBUT
7  i ← 1
8  TQ i ≤ n-1 FAIRE
9  imin ← i
10 j ← i+1
11 TQ j ≤ n FAIRE
12 SI T[j] < T[imin] ALORS
13 imin ← j
14 FSI
15 j ← j+1
16 FTQ
17 Swap(T, i, imin)
18 i ← i+1
19 FTQ
20 FIN
```

- ❖ Arrêt : deux boucles imbriquées à incrément constant
- ❖ Validité : $(T[1 : i - 1])$ est trié et $T[i - 1] \leq \min(T[i : n])$ est un invariant de boucle
- ❖ Complexité : $\Theta(n^2)$

Idée générale :

- ❏ Faire remonter les éléments les plus grands en échangeant les éléments contigus mal arrangés ;
- ❏ effectuer autant de passes que nécessaire pour que tous les éléments soient à la bonne place.

Tri par propagation (ou tri à bulles)

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)



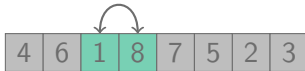
Tri par propagation (ou tri à bulles)

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)

4	6	1	8	7	5	2	3
---	---	---	---	---	---	---	---

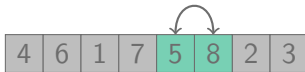
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)

4	6	1	7	8	5	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)

4	6	1	7	5	8	2	3
---	---	---	---	---	---	---	---

Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)

4	6	1	7	5	2	8	3
---	---	---	---	---	---	---	---

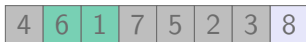
Tri par propagation (ou tri à bulles)



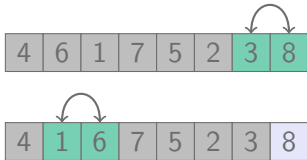
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



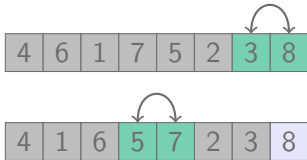
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



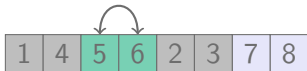
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



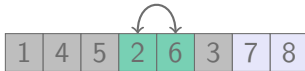
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



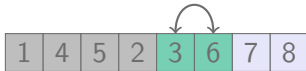
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



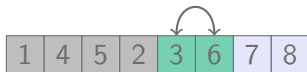
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



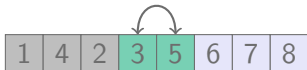
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



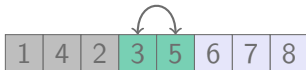
Tri par propagation (ou tri à bulles)



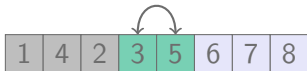
Tri par propagation (ou tri à bulles)



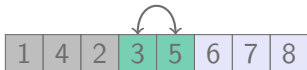
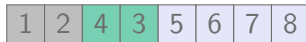
Tri par propagation (ou tri à bulles)



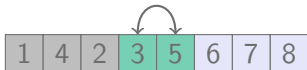
Tri par propagation (ou tri à bulles)



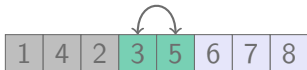
Tri par propagation (ou tri à bulles)



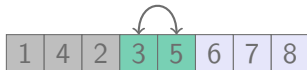
Tri par propagation (ou tri à bulles)



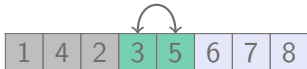
Tri par propagation (ou tri à bulles)



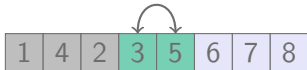
Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)



Tri par propagation (ou tri à bulles)

```
1 ALGORITHME TriBulles(T):
2 DONNEES
3   T: tableau d'entier de taille n
4 VARIABLES:
5   d, i: entiers
6 DEBUT
7   d ← n
8   TQ d > 1 FAIRE
9     i ← 1
10    TQ i < d FAIRE
11      SI T[i] > T[i+1] ALORS
12        Swap(T, i, i+1)
13      FSI
14      i ← i+1
15    FTQ
16  d ← d-1
17 FTQ
18 FIN
```

- ❖ Arrêt : deux boucles imbriquées à incrément constant
- ❖ Validité : ($T[d+1 : n]$ est trié et $t[d+1] \geq \max(T[1 : d])$) est un invariant
- ❖ Complexité : $\Theta(n^2)$

Tri par propagation (ou tri à bulles)

- ❑ Si aucun échange n'est effectué alors le tableau est trié.
- ❑ C'est un moyen simple pour savoir quand arrêter le travail de l'algorithme.
- ❑ On améliore ainsi la complexité dans le meilleur cas.

Tri par propagation (ou tri à bulles)

```
1  ALGORITHME TriBulles(T):
2  DEBUT
3    d ← n
4    echange ← VRAI
5    TQ echnage = VRAI FAIRE
6      i ← 1
7      echange ← FAUX
8      TQ i < d FAIRE
9        SI T[i] > T[i+1] ALORS
10         Swap(T, i, i+1)
11         echange ← VRAI
12      FSI
13      i ← i+1
14    FTQ
15    d ← d-1
16  FTQ
17  FIN
```

Tri par propagation (ou tri à bulles)

```
1  ALGORITHME TriBulles(T):  
2  DEBUT  
3    d ← n  
4    echange ← VRAI  
5    TQ echange = VRAI FAIRE  
6      i ← 1  
7      echange ← FAUX  
8      TQ i < d FAIRE  
9        SI T[i] > T[i+1] ALORS  
10         Swap(T, i, i+1)  
11         echange ← VRAI  
12      FSI  
13      i ← i+1  
14    FTQ  
15    d ← d-1  
16  FTQ  
17  FIN
```

Meilleur cas : $\Theta(n)$

Pire cas : $\Theta(n^2)$

Complexité : $O(n^2)$

Idée générale :

- Tri utiliser naturellement pour ranger des cartes ;
- on parcourt le tableau de droite à gauche et on range chaque élément à sa place parmi les éléments précédents.

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

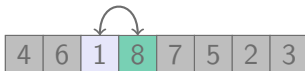
6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	1	8	7	5	2	3
---	---	---	---	---	---	---	---



Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	1	8	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

Diagram illustrating the insertion of 6 into the sorted subarray [4, 6, 8]. A curved arrow indicates the shift of 6 to the position before 8.

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	1	6	8	7	5	2	3
---	---	---	---	---	---	---	---

Diagram illustrating the insertion of 1 into the sorted subarray [4, 1, 6, 8]. A curved arrow indicates the shift of 6 to the position after 1.

Tri par insertion

6	4	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---



4	6	8	1	7	5	2	3
---	---	---	---	---	---	---	---

4	1	6	8	7	5	2	3
---	---	---	---	---	---	---	---

Tri par insertion



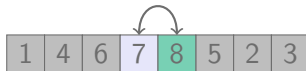
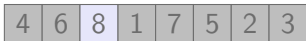
Tri par insertion



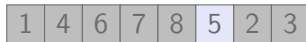
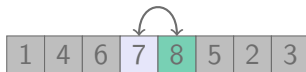
Tri par insertion



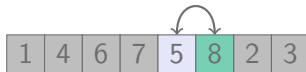
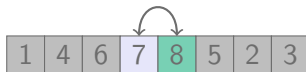
Tri par insertion



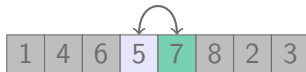
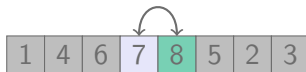
Tri par insertion



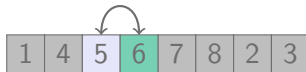
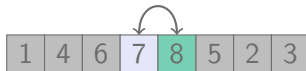
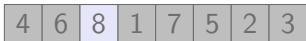
Tri par insertion



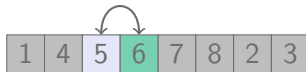
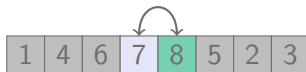
Tri par insertion



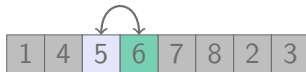
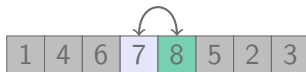
Tri par insertion



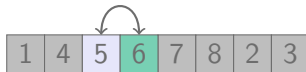
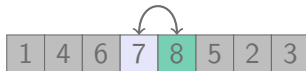
Tri par insertion



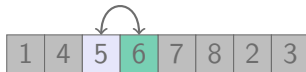
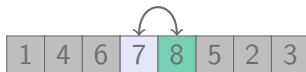
Tri par insertion



Tri par insertion



Tri par insertion



Tri par insertion

```
1  ALGORITHME TriInsertion(T):
2  DONNEES
3    T: tableau d'entier de taille n
4  VARIABLES:
5    i, j: entiers
6  DEBUT
7    i ← 2
8    TQ i ≤ n FAIRE
9      j ← i
10     TQ j > 1 ET T[j-1] > T[j] FAIRE
11       Swap(T, j, j-1)
12       j ← j-1
13     FTQ
14     i ← i+1
15   FTQ
16  FIN
```

```
1 ALGORITHME TriInsertion(T):  
2 DONNEES  
3   T: tableau d'entier de taille n  
4 VARIABLES:  
5   i, j: entiers  
6 DEBUT  
7   i ← 2  
8   TQ i ≤ n FAIRE  
9     j ← i  
10    TQ j > 1 ET T[j-1] > T[j] FAIRE  
11      Swap(T, j, j-1)  
12      j ← j-1  
13    FTQ  
14    i ← i+1  
15  FTQ  
16  FIN
```

- ❖ Arrêt : deux boucles imbriquées à incrément constant
- ❖ Validité : ($T[1 : i - 1]$ est trié) est un invariant
- ❖ Complexité : $O(n^2)$