

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

Algorithmes de tri (1)

Christian Lasou, Nour-Eddine Oussous, Éric Wegrzynowski

Licence ST-A, USTL - API1

12 février 2007

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

Tableaux triés

Définition (rappel)

Supposons le type E totalement ordonné par la relation notée \leq .
Un tableau $t[a..b]$ d'éléments de type E est trié pour l'ordre \leq si

$$\forall i \in \llbracket a, b-1 \rrbracket \quad t[i] \leq t[i+1]$$

Exemple

Exemple de tableau trié pour l'ordre usuel des caractères
($E = \text{CHAR}$)

1	2	3	4	5	6	7	8
E	I	L	M	N	O	O	T

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

Introduction

Tri par sélection

Principe

Algorithmes

Tri par insertion

Principe

Algorithme

Coût

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

Le problème

Problème du tri

Données $t[a..b]$ un tableau d'éléments de type E

But transformer t de sorte qu'il soit trié

Exemple

Transformer le tableau (de caractères)

T I M O L E O N

en

E I L M N O O T

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

TRI PAR SELECTION

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ●○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
Principe			

Les besoins

Il suffit donc de

- ▶ savoir déterminer l'indice du plus petit élément d'une tranche $t[a..b]$ de tableau ($\text{ind_min}(t, a, b)$)
- ▶ savoir échanger deux éléments d'un tableau ($\text{echanger}(x, y)$)

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ●○ ○○○○○	Tri par insertion ○○ ○○○○○ ○
------	--------------	----------------------------------	---------------------------------------

Principe

Principe du tri par sélection

Pour trier le tableau t

T I M O L E O N

- ▶ sélectionner l'élément le plus petit du tableau

T I M O L E O N

- ▶ l'échanger avec le premier élément du tableau

E I M O L T O N

- ▶ recommencer avec le reste du tableau

E I M O L T O N

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ●○○○○	Tri par insertion ○○ ○○○○○ ○
Algorithmes			

Sélection du minimum

$\text{ind_min}(t, a, b)$

Données : un tableau t , deux indices a, b

But : déterminer l'indice du plus petit élément de $t[a..b]$

Var. locales : k, i

```

i := a
{ t[i] ≤ t[a..a] }
pour k variant de succ(a) à b faire
    si t[k] < t[i] alors
        i := k
    fin si
    { t[i] ≤ t[a..k] }
fin pour
{ t[i] ≤ t[a..b] }
retourner i

```

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

ind_min en action

$t = \text{T I M O L E O N}$

$a = 1, b = 8$

k	i	min de $t[1..k-1]$
	1	T I M O L E O N
2	2	T I M O L E O N
3	2	T I M O L E O N
4	2	T I M O L E O N
5	2	T I M O L E O N
6	6	T I M O L E O N
7	6	T I M O L E O N
8	6	T I M O L E O N

```

i := a
pour k variant de succ(a) a b faire
  si t[k] < t[i] alors
    i := k
  fin si
fin pour
retourner i

```

Tri par sélection

tri_select (t)

Données : un tableau $t[a..b]$

But : trier le tableau t

Var. locales : k, i

```

{t[a..a-1] est trié et  $\forall j \in [a, a-1] \ t[j] \leq t[a..b]}$ 
pour k variant de a à b-1 faire
  i := ind_min(t, k, b)
  echanger(t[k], t[i])
  {t[a..k] est trié et  $\forall j \in [a, k] \ t[j] \leq t[k+1..b]}$ 
fin pour
{t[a..b-1] est trié et  $\forall j \in [a, b-1] \ t[j] \leq t[b]}$ 
{par conséquent t est trié}

```

Coût de la sélection du minimum

Pour un tableau t de taille n ($a = 1, b = n$) :

► Nombre de comparaisons :

$$c_n = \sum_{k=2}^n 1 = n - 1 \sim n$$

tri_select en action

$t = \text{T I M O L E O N}$

k	i	actions	t
1	ind_min(t, 1, 8) = 6	echanger(t [1], t [6])	E I M O L T O N
2	ind_min(t, 2, 8) = 2	echanger(t [2], t [2])	E I M O L T O N
3	ind_min(t, 3, 8) = 5	echanger(t [3], t [5])	E I L O M T O N
4	ind_min(t, 4, 8) = 5	echanger(t [4], t [5])	E I L M O T O N
5	ind_min(t, 5, 8) = 8	echanger(t [5], t [8])	E I L M N T O O
6	ind_min(t, 6, 8) = 7	echanger(t [6], t [7])	E I L M N O T O
7	ind_min(t, 7, 8) = 8	echanger(t [7], t [8])	E I L M N O O T

Plan	Introduction	Tri par sélection ○○ ○○○○●	Tri par insertion ○○ ○○○○○○ ○
Algorithmes			

Coût du tri par sélection

Pour un tableau t de taille n ($a = 1$, $b = n$) :

► Nombre d'échanges :

$$e_n = \sum_{i=1}^{n-1} 1 = n - 1 \sim n$$

► Nombre de comparaisons :

$$c_n = \sum_{k=1}^{n-1} (n - k) = \frac{n(n-1)}{2} \sim \frac{n^2}{2}$$

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ●○ ○○○○○ ○
Principe			

Principe du tri par insertion

Pour trier le tableau t

T I M O L E O N

1. on part du constat que la tranche $t[1..1]$ est triée

T I M O L E O N

2. puis on met l'élément d'indice 2 à sa place dans la tranche $t[1..2]$

I T M O L E O N

3. la tranche $t[1..2]$ étant maintenant triée, on recommence avec l'étape 2 avec l'élément suivant du tableau.

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○○○○ ○

TRI PAR INSERTION

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ●○ ○○○○○ ○
Principe			

Les besoins

Il suffit donc de

- savoir insérer l'élément d'indice k d'un tableau t dans la tranche $t[a..k]$, $t[a..k-1]$ étant supposé trié (**insérer** (t, k)), de sorte que $t[a..k]$ soit une tranche triée.

Algorithmes de tri (1)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ●○○○○○ ○
Algorithme			

Insertion dans un tableau trié

insérer(t, k)

Données : un tableau $t[a..b]$, un indice $k > a$

CU : la tranche $t[a..k-1]$ est triée

But : placer l'élément $t[k]$ à sa place dans la tranche $t[a..k]$

Var. locales : i

```

i := k
{ t[a..i-1] trié, t[i] < t[i+1..k], t[i+1..k] trié }
tant que i > a et t[i] < t[i-1] faire
    echanger(t[i], t[i-1])
    dec(i)
{ t[a..i-1] trié, t[i] < t[i+1..k], t[i+1..k] trié }
fin tant que
{ t[a..i-1] trié, t[i] < t[i+1..k], t[i+1..k] trié
  et (i = a ou t[i] ≥ t[i-1]) }

```

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○●○○○ ○
Algorithme			

insérer en action (1)

$t = \text{I L M O T E O N}$
 $k = 6$

i		t
6	$t[i] < t[i-1]$	I L M O E T O N
5	$t[i] < t[i-1]$	I L M E O T O N
4	$t[i] < t[i-1]$	I L E M O T O N
3	$t[i] < t[i-1]$	I E L M O T O N
2	$t[i] < t[i-1]$	E I L M O T O N
1	$i = a$	E I L M O T O N

$i := k$
tant que $i > a$ et $t[i] < t[i-1]$ **faire**
 echanger($t[i], t[i-1]$)
 dec(i)
fin tant que

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ●○○○○○ ○
Algorithme			

insérer en action (1)

$t = \text{I M O T L E O N}$
 $k = 5$

i		t
5	$t[i] < t[i-1]$	I M O L T E O N
4	$t[i] < t[i-1]$	I M L O T E O N
3	$t[i] < t[i-1]$	I L M O T E O N
2	$t[i] \geq t[i-1]$	I L M O T E O N

$i := k$
tant que $i > a$ et $t[i] < t[i-1]$ **faire**
 echanger($t[i], t[i-1]$)
 dec(i)
fin tant que

Plan	Introduction	Tri par sélection ○○ ○○○○○	Tri par insertion ○○ ○○●○○○ ○
Algorithme			

Coût de l'insertion

Pour insérer l'élément d'indice n dans la tranche $t[1..n-1]$:

- ▶ Meilleur des cas : $t[n] \geq t[1..n-1]$ (le tableau est trié)

- ▶ nbre échanges :

$$e_n = 0$$

- ▶ nbre comparaisons :

$$c_n = 1$$

- ▶ Pire des cas : $t[n] < t[1..n-1]$

- ▶ nbre échanges :

$$e_n = n - 1 \sim n$$

- ▶ nbre comparaisons :

$$c_n = n - 1 \sim n$$

Tri par insertion

tri_insert (t)

Données : un tableau $t[a..b]$

But : trier le tableau t

Var. locales : k

{t[1..1] est trié}
pour k variant de succ(a) à b faire
 insérer(t,k)
 {t[a..k] est trié}
fin pour
{t[a..b] est trié}

Coût du tri par insertion

Pour un tableau t de taille n :

- ▶ Meilleur des cas : $\forall k > a \ t[k] \geq t[a..k - 1]$ le tableau est trié
 - ▶ nbre échanges : $e_n = \sum_{k=2}^n 0 = 0$
 - ▶ nbre comparaisons : $c_n = \sum_{k=2}^n 1 = n - 1 \sim n$
- ▶ Pire des cas : $\forall k > a \ t[k] < t[a..k - 1]$ le tableau est trié dans l'ordre inverse
 - ▶ nbre échanges : $e_n = \sum_{k=2}^n (k - 1) = \frac{n(n-1)}{2} \sim n^2$
 - ▶ nbre comparaisons : $c_n = \sum_{k=2}^n (k - 1) = \frac{n(n-1)}{2} \sim n^2$

tri_insert en action

t = T I M O L E O N

k	actions	t
2	insérer (t,2)	I T M O L E O N
3	insérer (t,3)	I M T O L E O N
4	insérer (t,4)	I M O T L E O N
5	insérer (t,5)	I L M O T E O N
6	insérer (t,6)	E I L M O T O N
7	insérer (t,7)	E I L M O O T N
8	insérer (t,8)	E I L M N O O T

pour k variant de succ(a)
 a b faire
 insérer(t,k)
fin pour