

Plan	Tri par fusion ○○ ○○ ○○○○	Comparaison
------	------------------------------------	-------------

Plan	Tri par fusion ○○ ○○ ○○○○	Comparaison
------	------------------------------------	-------------

Algorithmes de tri (2)

Ch. Lasou, N.E. Oussous, E. Wegrzynowski

Licence ST-A, USTL - API1

19 février 2007

Algorithmes de tri (2)	Licence ST-A, USTL - API1
------------------------	---------------------------

Algorithmes de tri (2)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Tri par fusion ●○○ ○○ ○○○○	Comparaison
Principe		

Principe du tri par fusion

Pour trier le tableau t

$t[1..8] = \boxed{T} \boxed{I} \boxed{M} \boxed{O} \boxed{L} \boxed{E} \boxed{O} \boxed{N}$

- ▶ on coupe le tableau en deux parties de même taille

$t[1..4] = \boxed{T} \boxed{I} \boxed{M} \boxed{O}$ et $t[5..8] = \boxed{L} \boxed{E} \boxed{O} \boxed{N}$

- ▶ on trie chacune des deux moitiés

$t[1..4] = \boxed{I} \boxed{M} \boxed{O} \boxed{T}$, $t[5..8] = \boxed{E} \boxed{L} \boxed{N} \boxed{O}$

- ▶ puis on fusionne ces deux tableaux triés

$t[1..8] = \boxed{E} \boxed{I} \boxed{L} \boxed{M} \boxed{N} \boxed{O} \boxed{O} \boxed{T}$

Algorithmes de tri (2)	Licence ST-A, USTL - API1
------------------------	---------------------------

Plan	Tri par fusion ●○○ ○○ ○○○○	Comparaison
Principe		

Les besoins

Il suffit donc de

- ▶ savoir fusionner deux tranches triées $t[a..b]$ et $t[b+1..c]$ de façon à obtenir une tranche $t[a..c]$ triée.

- ▶ Spécification de la procédure de fusion

fusion(t, a, b, c)

Données : un tableau t , trois indices a, b, c

CU : les tranches $t[a..b]$ et $t[b+1..c]$ sont triées

But : rendre triée la tranche $t[a..c]$

Algorithmes de tri (2)	Licence ST-A, USTL - API1
------------------------	---------------------------

Algorithme récursif

- l'algorithme de tri par fusion est **récursif**
- la récursion dépendant des indices délimitant les tranches

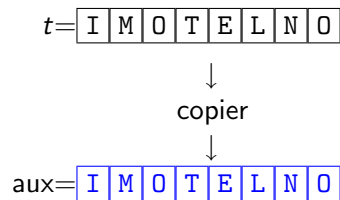
- spécification de la procédure

tri_fusion_rec (*t*, *a*, *b*)

Données : un tableau *t*, deux indices *a* et *b*

But : trier la tranche *t[a..b]*

Fusion en action



<i>k</i>	<i>i</i>	<i>j</i>	<i>t</i>							
	1	5	I	M	O	T	E	L	N	O
1	1	6	E	M	O	T	E	L	N	O
2	2	6	E	I	O	T	E	L	N	O
3	2	7	E	I	L	T	E	L	N	O
4	3	7	E	I	L	M	E	L	N	O
5	3	8	E	I	L	M	N	L	N	O
6	4	8	E	I	L	M	N	O	N	O
7		9	E	I	L	M	N	O	O	O
8	4		E	I	L	M	N	O	O	T

Algorithme de la fusion

fusion (*t*, *a*, *b*, *c*)

Données : un tableau *t*, trois indices *a*, *b*, *c*

CU : les tranches *t[a..b]* et *t[b + 1..c]* sont triées

But : rendre triée la tranche *t[a..c]*

```

copier  t[a..c] dans aux[a..c]

i := a  j := succ(b)
pour k := a à c faire
  si (i ≤ b) et (j ≤ c) alors
    si aux[i] ≤ aux[j] alors
      t[k] := aux[i] inc(i)
    sinon
      t[k] := aux[j] inc(j)
  sinon si i ≤ b alors
    t[k] := aux[i] inc(i)
  sinon
    t[k] := aux[j] inc(j)
fin si
fin pour
    
```

Coût de la fusion

Pour fusionner *t[a..b]* de taille *n*₁ avec *t[b + 1..c]* de taille *n*₂

(*n* = *n*₁ + *n*₂)

- Dans tous les cas le nombre d'affectations est :

$$a_n = 2n$$

- Nombre de comparaisons

- Meilleur des cas : tous les éléments de la tranche la plus petite sont inférieurs ou égaux à ceux de l'autre tranche

$$c_n = \min(n_1, n_2)$$

- Pire des cas : les deux éléments les plus grands de la tranche *t[a..c]* se trouvent dans deux tranches différentes

$$c_n = n - 1$$

Algorithme du tri

tri_fusion_rec (t,a,b)

Données : un tableau t , deux indices a et b

But : trier la tranche $t[a..b]$

```

si a < b alors
  m := milieu(a,b)
  tri_fusion_rec(t,a,m)
  tri_fusion_rec(t,succ(m),b)
  fusion(t,a,m,b)
fin si
    
```

Coût du tri

Nombre a_n d'affectations pour trier un tableau de taille n

► défini de manière récurrente par

$$a_1 = 0$$

$$a_n = a_{\lfloor n/2 \rfloor} + a_{\lceil n/2 \rceil} + 2n \quad \forall n > 1$$

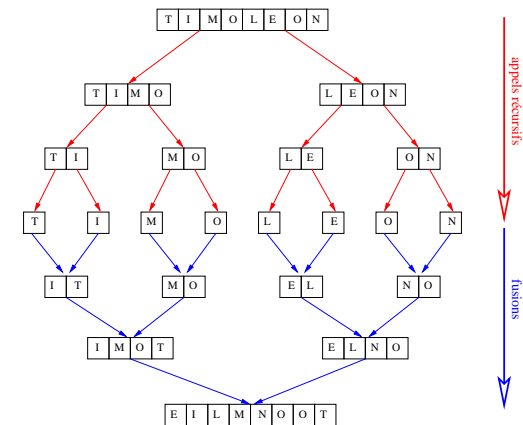
► lorsque $n = 2^p$ est une puissance de deux, on trouve

$$a_n = 2n \log_2(n)$$

► dans le cas général

$$a_n = \theta(n \log_2(n))$$

Exemple



Coût du tri

Nombre c_n de comparaisons pour trier un tableau de taille n

► Relation de récurrence

$$c_1 = 0$$

$$c_n = c_{\lfloor n/2 \rfloor} + c_{\lceil n/2 \rceil} + c'_n \quad \forall n > 1$$

où c'_n = nbre de comparaisons dans la fusion

► Dans le meilleur des cas : $c'_n = \lfloor n/2 \rfloor$

► Dans le pire des cas : $c'_n = n - 1$

► Dans tous les cas

$$c_n = \theta(n \log_2(n))$$

Ordre de grandeur du nombre de comparaisons

	Tri par sélection	Tri par insertion	Tri par fusion
Meilleur des cas	$\sim \frac{n^2}{2}$	$\sim n$	$\theta(n \log_2 n)$
Pire des cas	$\sim \frac{n^2}{2}$	$\sim \frac{n^2}{2}$	$\theta(n \log_2 n)$

Nombre de comparaisons pour trier des tableaux de taille ≤ 200

