

Tri externe

UMLV ©

Méthodes de classement destinées à de grandes masses de données

Applicables à des fichiers séquentiels

Complexité : évaluée surtout en nombre d'opérations entrées/sorties (lectures/écritures sur fichier).

326

Partage/Fusion

UMLV ©

Pour tri externe :
Partage élémentaire

bla bla bla bla bla

PARTAGE

bla bla bl a bla bla

tri tri
a a b b b ! ! ! a a b b ! !

Fusion essentielle

FUSION

a a a a b b b b b ! ! ! ! !

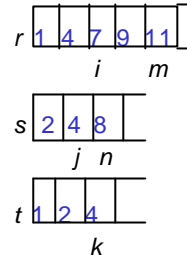
Sur fichier séquentiel : méthode itérative.

327

Fusion

UMLV ©

$L_1 = (1, 4, 7, 9, 11)$
 $L_2 = (2, 4, 8)$
 Fusion (L_1, L_2) = $(1, 2, 4, 4, 7, 8, 9, 11)$



```

i ← 1; j ← 1;
r[m+1] ← ∞; s[n+1] ← ∞;
pour k ← 1 à m+n faire
  si r[i] ≤ s[j] alors
    { t[k] ← r[i]; i ← i+1; }
  sinon
    { t[k] ← s[j]; j ← j+1; }
    
```

328

```

fonction TRI ( S : suite ) : suite ;
début
  si | S | ≤ 1 alors
    retour S;
  sinon début
    (S1, S2) ← PARTAGE( S );
    T1 ← TRI(S1); T2 ← TRI(S2);
    retour FUSION(T1, T2);
  fin
fin
    
```

Temps de calcul, si PARTAGE et FUSION linéaires :

$$T(n) = 2T(n/2) + an + \beta, \quad n > 1$$

$$T(0) = T(1) = ?$$

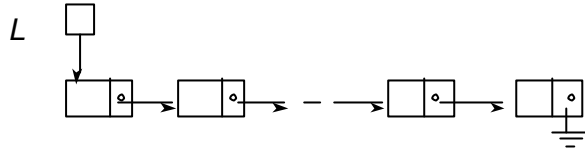
$$\Rightarrow T(n) = ? (n \log n)$$

UMLV ©

329

Tri de listes chaînées

UMLV ©



```

fonction TRI (L : liste) : liste ;
début
  si L = NULL ou L->suivant = NULL alors
    retour (L)
  sinon début
    P ← MILIEU (L) ;
    M ← P->suivant ; P->suivant ← NULL ;
    retour (FUSION(TRI(P),TRI(M)));
  fin
fin
  
```

330

Milieu

UMLV ©



```

fonction MILIEU (L liste) : pointeur ;
  /* L contient au moins 2 éléments */
début
  p ← L ; q ← L->suivant ;
  tant que q ≠ NULL faire début
    q ← q->suivant ;
    si q ≠ NULL alors
      { q ← q->suivant ; p ← p->suivant ; }
  fin
  retour p ;
fin
  
```

331

EXERCICE : écrire la fusion de listes classées

TEMPS

MILIEU (L) \longrightarrow $O(|L|)$

FUSION (L, M) \longrightarrow $O(|L| + |M|)$

\Rightarrow TRI(L) \longrightarrow $O(|L| \log |L|)$

Espace

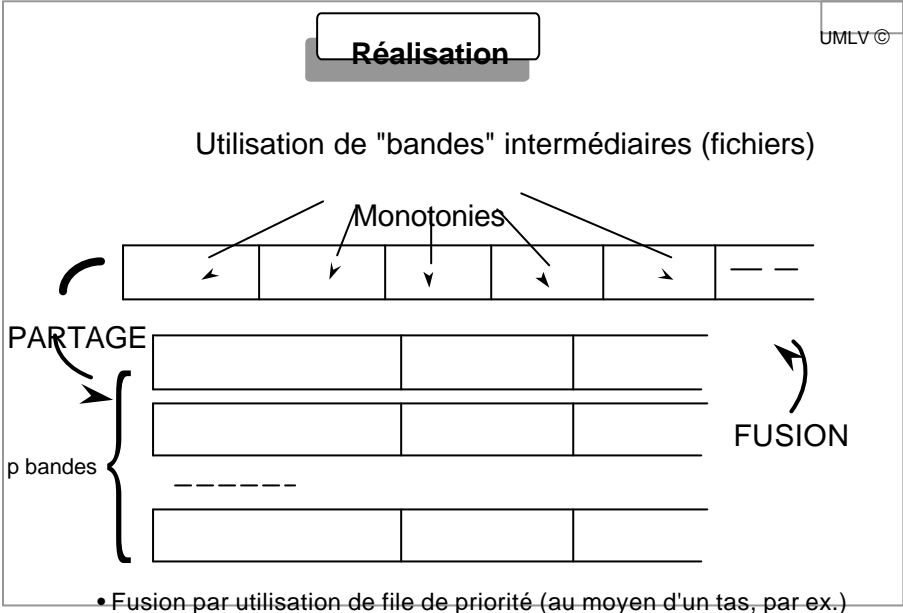
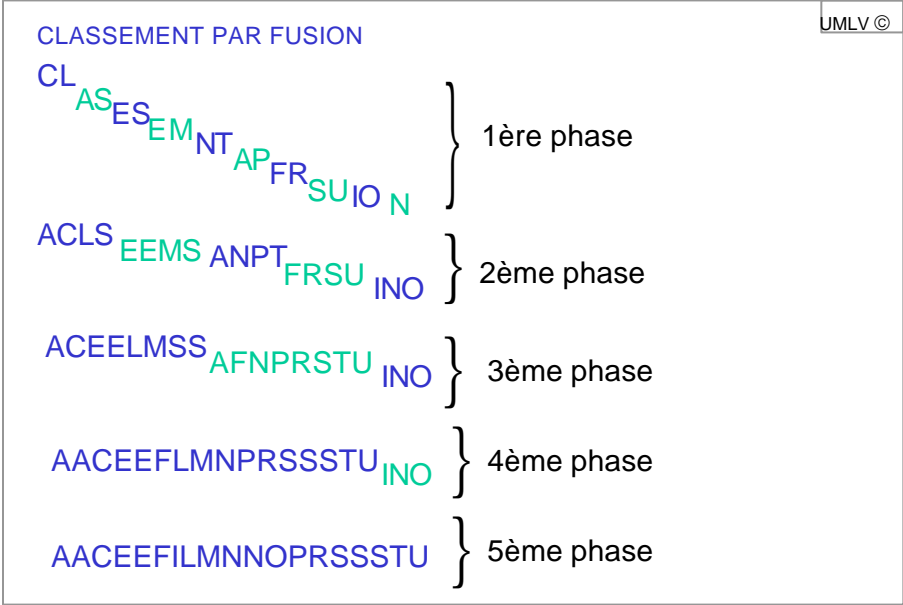
$O(L)$ pour la pile de récursion

Propriété

TRI stable (si la fusion bien écrite) :
l'ordre relatif d'éléments de même clé
est inchangé

Classement de fichiers

- Fichiers séquentiels
- **Méthode stupide** (?) :
 - 1- transférer le fichier en mémoire centrale
 - 2- appliquer un tri interne
 - avec utilisation de mémoire virtuelle (disque)
 Méthode NON-OPTIMISEE !
- **Méthode par fusions** :
 - 1- création de monotonies (sous-suites croissantes)
 - 2- répartition des monotonies
 - 3- fusion des monotonies



Tri équilibré

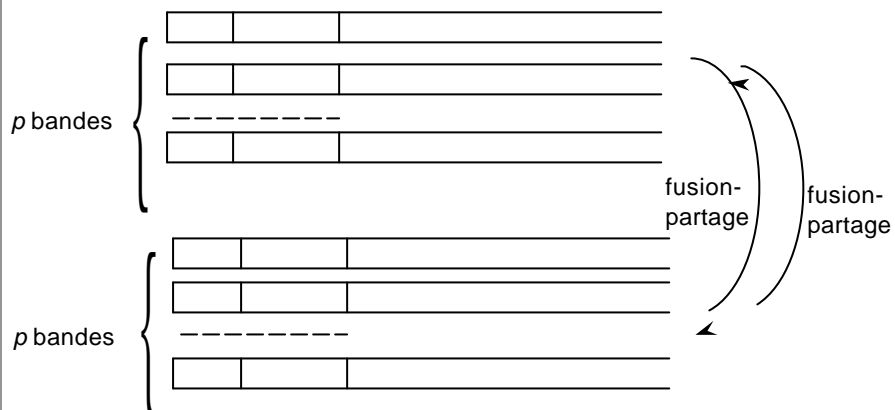
UMLV ©

- environ autant de monotonies sur chacune des p bandes
- monotonies de même longueur (sauf la dernière) ou monotonies séparées
- longueurs successives : $1, p, p^2, \dots$
- $\log_p n$ phases
- $2n \log_p n$ lectures et écritures d'éléments
- si monotonies initiales de longueur m : $\log_p(n/m)$ phases

336

- $2p$ bandes intermédiaires
- $n(2 + \log_p n)$ lectures et écritures

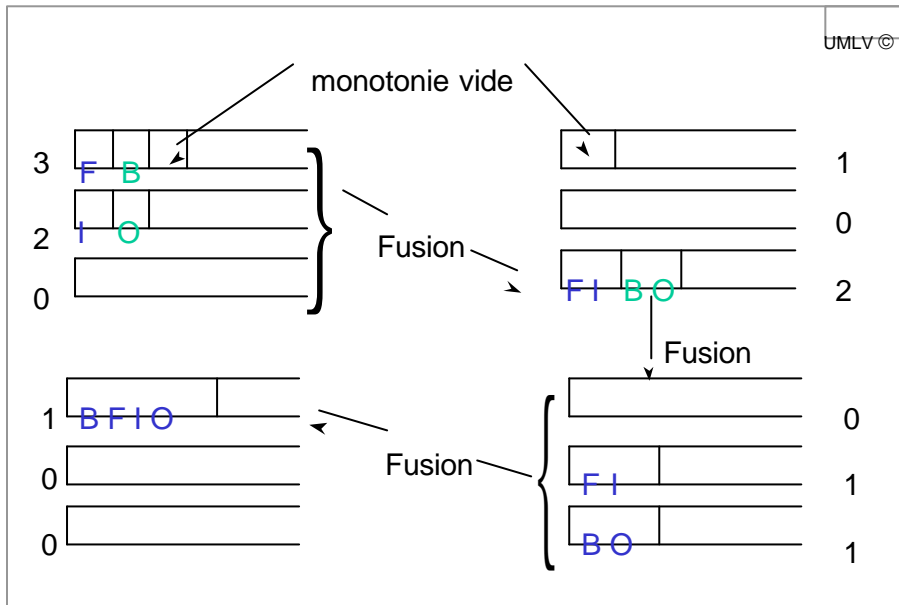
UMLV ©



337

Tri polyphase

- répartition particulière des monotonies pour une meilleure utilisation des bandes
- Exemple : répartition de Fibonacci avec 3 bandes
 - $\rightarrow \lfloor \log_f(n) \rfloor + 1$ phases
 - $< n(3 + \lfloor \log_f(n) \rfloor)$ lectures et écritures
- Introduction de monotonies vides pour répartition idéale



Répartitions

UMLV ©

- avec 3 bandes

bande 1 : 1 0 1 3 0 5 13 ...

bande 2 : 0 1 0 2 5 0 8

bande 3 : 0 1 2 0 3 8 0

- avec 5 bandes

bande 1 : 1 0 1 3 7 15 0 ...

bande 2 : 0 1 0 2 6 14 29

bande 3 : 0 1 2 0 4 12 27

bande 4 : 0 1 2 4 0 8 23

bande 5 : 0 1 2 4 8 0 15

340

Construction de monotonies

UMLV ©

- de même longueur : par tri interne
- de longueur variable : **sélection et remplacement**
utilisation d'une file de priorité représentée au moyen d'un tas

Propriété : si l'ordre des éléments est aléatoire et la file de taille m ,
la longueur moyenne des monotonies est $2m$.

341

Sélection - Remplacement

UMLV ©

file initiale x

tables en mémoire centrale

min

f

file de priorité

liste d'attente

monotonies ...d

gestion {

- si $x < d$, x dans liste d'attente
- si $d \leq x \leq f$, x dans monotonie
- si $f \leq x$, f dans monotonie et x dans file de priorité

invariant
 $d \leq f$