Les méthodes de tri externes (suite)

Les algorithmes de fusion

- Les algorithmes de tri externes sont basés sur la fusion de fichiers ordonnés.

- La fusion de deux (ou plus) fichiers ordonnés consiste à produire un fichier ordonné comportant l'union des enregistrements des fichiers de départ.

- La fusion peut se réaliser avec une complexité linéaire en fonction de la taille des fichiers à fusionner.

Principe de l'algorithme de fusion

- on compare les deux enregistrements courants,
- le plus petit de ces deux enregistrements est ajouté en fin du fichier résultat,
- on passe à l'enregistrement suivant dans le fichier qui contenait le plus petit des deux enregistrements.

Le processus est répété jusqu'à ce que l'on atteigne la fin d'un des deux fichiers. Il suffit ensuite de copier le reste de l'autre fichier en fin du fichier résultat.

Le concept d'arbre de sélection

- Si l'on fusionne M fichiers, a chaque élément lu il est nécessaire de faire $M - 1$ comparaisons.

- Ces comparaisons sont en bonne partie redondantes et il est possible de les éviter à l'aide d'une structure de données appelée *arbre de sélection*.

- Un arbre de sélection est un arbre binaire dont chaque noeud comporte la valeur du plus petit de ses deux successeurs.

- Lorsque l'on change la valeur d'une des M feuilles d'un arbre de sélection, la mise à jour de l'arbre ne nécessite que $\log(M)$ comparaisons.

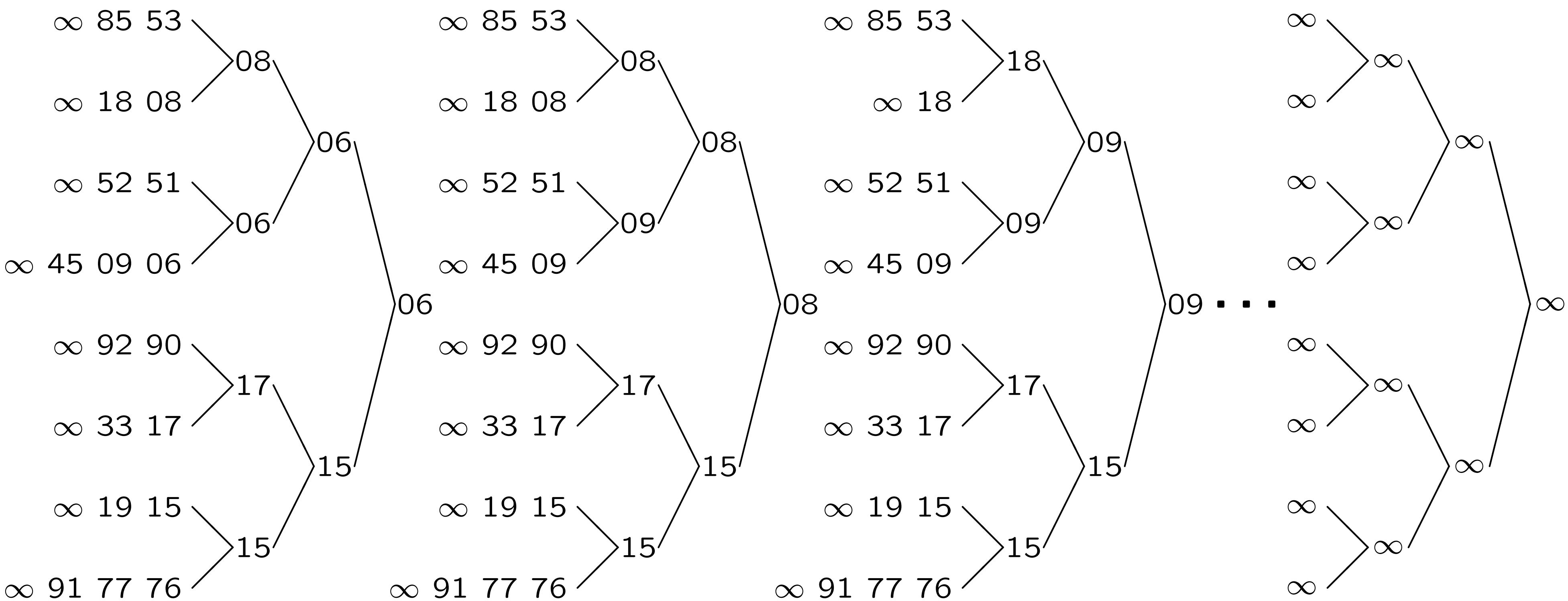
Fusion avec arbre de sélection

Si on donne aux feuilles les valeurs des premiers enregistrements de M fichiers à fusionner, une étape de l'algorithme de fusion consiste à

- ajouter l'enregistrement désigné par la racine de l'arbre de sélection en fin du fichier résultat,
- remplacer la valeur de la feuille correspondante par l'enregistrement suivant dans le fichier qui lui est associé,
- mettre à jour l'arbre de sélection.

La mise à jour de l'arbre consiste à modifier la valeur des noeuds situés sur la branche joignant la feuille à la racine. Il ne faudra donc effectuer que $\lceil \log_2 M \rceil$ comparaisons

Exemple :



∞ 85 53
∞ 18 08
∞ 52 51
∞ 45 09 06
∞ 92 90
∞ 33 17
∞ 19 15
∞ 91 77 76
08
06
17
15
06
15
06
∞ 85 53
∞ 18 08
∞ 52 51
∞ 45 09
∞ 92 90
∞ 33 17
∞ 19 15
∞ 91 77 76
08
09
17
15
08
15
08
∞ 85 53
∞ 18
∞ 52 51
∞ 45 09
∞ 92 90
∞ 33 17
∞ 19 15
∞ 91 77 76
18
09
17
15
09
15
09 . . .
∞

Le tri par fusion : point de départ

- Le point de départ est un fichier accessible séquentiellement.
- On dispose de la possibilité d'utiliser un nombre m de fichiers auxiliaires.
- Les enregistrements du fichier de départ peuvent se présenter sous la forme de groupe d'enregistrements ou *séries* triées soit présentes naturellement, soit obtenues par un tri en mémoire centrale.

Le tri par fusion équilibré à n voies

On dispose de $2n$ fichiers auxiliaires répartis en 2 groupes de n.
L'algorithme comporte deux phases.

1. *La phase d'initialisation.*
   Les séries triées initiales de longueur ℓ sont réparties entre les n premiers fichiers auxiliaires. Appelons ceux-ci les *fichiers d'entrée.*

2. *La phase de fusion.*
   Cette phase nécessite généralement plusieurs passes. Lors d'une passe i :

   (a) On fusionne chaque groupe de n séries de longueur $\ell n^{(i-1)}$ constitué des séries de position identique dans chaque fichier d'entrée. On répartit les séries résultantes de longueur ℓn^i entre les n autres fichiers auxiliaires (*les fichiers de sortie*).

   (b) Dans le cas où tous les fichiers d'entrée n'ont pas le même nombre de séries, on ignore les séries manquantes.

   Lors de chaque nouvelle passe, on inverse le rôle des fichiers d'entrée et de sortie.

3. Le tri s'achève lorsque l'on n'obtient plus qu'une seule série.

Exemple : 10 séries à trier, 4 fichiers auxiliaires.

Passes	Fichiers	Séries
0 (Initialisation)	A_1	s_1 s_3 s_5 s_7 s_9
	A_2	s_2 s_4 s_6 s_8 s_{10}
	A_3	—
	A_4	—
1	A_1	—
	A_2	—
	A_3	$s_{(1,2)}$ $s_{(5,6)}$ $s_{(9,10)}$
	A_4	$s_{(3,4)}$ $s_{(7,8)}$
2	A_1	$s_{(1,2,3,4)}$ $s_{(9,10)}$
	A_2	$s_{(5,6,7,8)}$
	A_3	—
	A_4	—
3	A_1	—
	A_2	—
	A_3	$s_{(1,2,3,4,5,6,7,8)}$
	A_4	$s_{(9,10)}$
4	A_1	$s_{(1,2,3,4,5,6,7,8,9,10)}$
	A_2	—
	A_3	—
	A_4	—

Le tri non équilibré

- On dispose de t fichiers auxilliaires.

- On réparti les t fichiers auxilliaires en 2 groupes respectivement de p et $t - p$ fichiers.

Exemple : 10 séries et $5 = 3 + 2$ fichiers auxilliaires.

Passes	Fichiers	Séries
0 (initialisation)	A_1	s_1 s_4 s_7 s_{10}
	A_2	s_2 s_5 s_8
	A_3	s_3 s_6 s_9
	A_4	—
	A_5	—
1	A_1	—
	A_2	—
	A_3	—
	A_4	$s_{(1,2,3)}$ $s_{(7,8,9)}$
	A_5	$s_{(4,5,6)}$ s_{10}
2	A_1	$s_{(1,2,3,4,5,6)}$
	A_2	$s_{(7,8,9,10)}$
	A_3	—
	A_4	—
	A_5	—
3	A_1	—
	A_2	—
	A_3	—
	A_4	$s_{(1,2,3,4,5,6,7,8,9,10)}$
	A_5	—

Le tri par fusion : analyse

- Le but est de calculer le nombre de passes de fusion nécessaires lorsque l'on utilise deux groupes de p et de $(t-p)$ fichiers auxiliaires.

- Soit e le nombre total d'enregistrements à trier et ℓ la longueur des séries.

- Le nombre initial de séries s est donné par :

$$s = \left\lceil \frac{e}{\ell} \right\rceil .$$

Le tableau ci-dessous indique le nombre de passes de fusion nécessaires en fonction de ce nombre initial de séries.

Nombre de passes	Nombre initial de séries	
1	$1 < s \leq p$	
2	$p < s \leq p(t-p)$	
3	$p(t-p) < s \leq p^2(t-p)$	
4	$p^2(t-p) < s \leq p^2(t-p)^2$	
$\vdots$	$\vdots$	
$2k$	$p^k(t-p)^{k-1} < s \leq p^k(t-p)^k$	(1)
$2k+1$	$p^k(t-p)^k < s \leq p^{k+1}(t-p)^k$	(2)

Analyse : Tri équilibré à p voies

- Soit $p = t/2$ et soit n le nombre de passes nécessaires.

- On a donc $n = 2k$ dans le cas pair (équation (1)) et $n = 2k + 1$ dans le cas impair (équation (2)).

- En remplaçant k par $\frac{n}{2}$ dans (1) et par $\frac{n-1}{2}$ dans (2), ces équations deviennent :

$$p^{n-1} < s \leq p^n.$$

- Sachant que $\log a^b = b \log a$, cette dernière inégalité peut se réécrire

$$(n-1)\log p < \log s \leq n \log p$$

ou encore

$$(n-1) < \frac{\log s}{\log p} \leq n.$$

- D'où, sachant que $\frac{\log a}{\log b} = \log_b a$,

$$(n-1) < \log_p s \leq n.$$

Dès lors, n le nombre total de passes de fusion pour le tri équilibré à p voies vaut

$$\left\lceil \log_p s \right\rceil.$$

Analyse : Tri non équilibré

- Supposons, par exemple, que le nombre de passes n est impair ($n = 2k + 1$).

- En remplaçant k par $\frac{n-1}{2}$ dans l'équation (2), celle-ci s'écrit :

$$p^{\frac{n-1}{2}}(t-p)^{\frac{n-1}{2}} < s \leq p^{\frac{n+1}{2}}(t-p)^{\frac{n-1}{2}}$$

ou encore, étant donné que $\log ab = \log a + \log b$:

$$(n-1)\log\sqrt{p(t-p)} < \log s \leq (n-1)\log\sqrt{p(t-p)} + \log p.$$

- En divisant par $\log \sqrt{p(t-p)}$, on a :

$$(n-1) < \frac{\log s}{\log \sqrt{p(t-p)}} \leq (n-1) + \frac{\log p}{\log \sqrt{p(t-p)}}.$$

- Vu que pour $p \in [2, t-2]$, on a

$$0 < \frac{\log p}{\log \sqrt{p(t-p)}} < 2,$$

on peut dire qu'en général le nombre de passes nécessaires vaut :

$$\left\lceil \frac{\log s}{\log \sqrt{p(t-p)}} \right\rceil.$$

- Par un raisonnement similaire, on obtient le même résultat lorsque n est pair.

Analyse : valeur optimale de p

- Essayons de déterminer la valeur optimale de p qui permet, étant donné un nombre fixé de passes, de trier un nombre maximum de séries.

- Soit le cas où le nombre de passes n est pair ($n = 2k$).

- L'équation (1) indique que le nombre maximum de séries est :

$$p^k(t-p)^k$$

- En dérivant par rapport à p, il vient

$$kp^{k-1}(t-p)^k - p^k k(t-p)^{k-1} = 0$$

ou encore

$$kp^{k-1}(t-p)^{k-1}\,(t-2p) = 0.$$

- Ainsi, le choix optimal pour p est :

$$p = \left\lceil \frac{t}{2} \right\rceil .$$

Le tri polyphase

Le tri polyphase : motivation

- Dans le cas de 3 fichiers auxiliaires, une passe sur deux est une passe sans fusion.

- Une passe sans fusion est un simple recopiage et ne fait pas vraiment progresser le tri.

- Il est possible d'éviter les recopiages sans fusion en utilisant une répartition non uniforme des séries lors de l'initialisation : c'est la technique du tri *polyphase*.

Motivation du tri polyphase : exemple

Passes	Fichiers	Séries
0 (Initialisation)	A_1	s_1 s_3 s_5 s_7 s_9
	A_2	s_2 s_4 s_6 s_8 s_{10}
	A_3	$-$
	A_4	$-$
1	A_1	$-$
	A_2	$-$
	A_3	$s_{(1,2)}$ $s_{(5,6)}$ $s_{(9,10)}$
	A_4	$s_{(3,4)}$ $s_{(7,8)}$
2	A_1	$s_{(1,2,3,4)}$
	A_2	$s_{(5,6,7,8)}$
	A_3	$s_{(9,10)}$
	A_4	$-$
3	A_1	$-$
	A_2	$-$
	A_3	$-$
	A_4	$s_{(1,2,3,4,5,6,7,8,9,10)}$

Le tri polyphase : algorithme général

1. On distribue les séries dans $t-1$ fichiers d'entrée suivant une partition a déterminer.

2. Une passe de fusion consiste à fusionner les séries successives se trouvant dans les fichiers d'entrée et à inscrire les séries résultantes dans le fichier de sortie.

3. Chaque passe se termine lorsqu'un des fichiers d'entrée a été intégralement utilisé. Ce fichier devient alors le fichier de sortie pour la passe suivante et l'ancien fichier de sortie devient quant à lui un fichier d'entrée.

4. Le problèm est de bien choisir la distribution initiale.

Le tri polyphase : Calcul de la distribution initiale optimale (3 fichiers auxiliaires)

n	A_1	A_2	A_3	total
0	1	0	0	1
1	0	1	1	2
2	1	0	2	3
3	3	2	0	5
4	0	5	3	8
5	5	0	8	13
6	13	8	0	21