

Cours de bases de données,
aspects systèmes,
<http://sys.bdpedia.fr>

Algorithme de tri

Tri externe

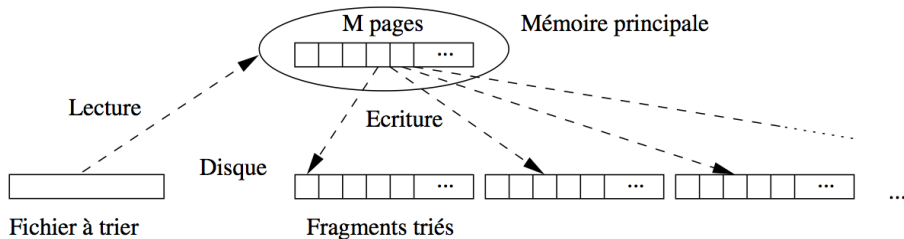
Le **tri externe** est utilisé,

- pour les algorithmes de jointure (*sort/merge*)
- l'élimination des doublons (*clause distinct*)
- pour les opérations de regroupement (*group by*)
- ... et bien sûr pour les *order by*

C'est une opération qui peut être très coûteuse sur de grands jeux de données.

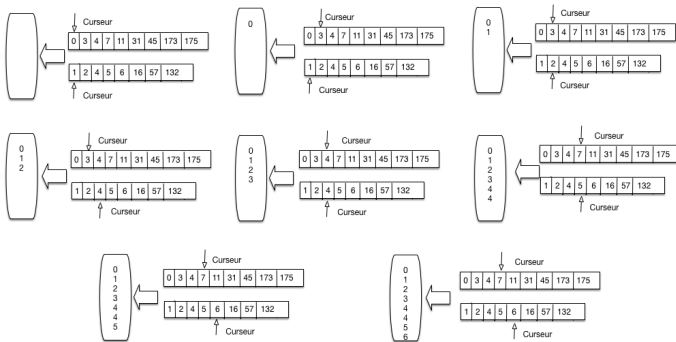
Première phase : le tri

On remplit la mémoire, on trie, on vide dans des **fragments**, et on recommence.



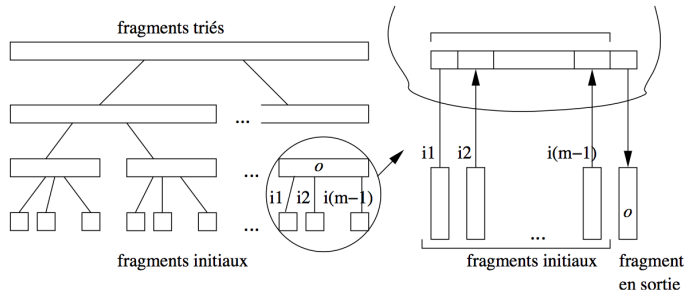
Coût : une lecture + une écriture du fichier.

Fusion de listes triées



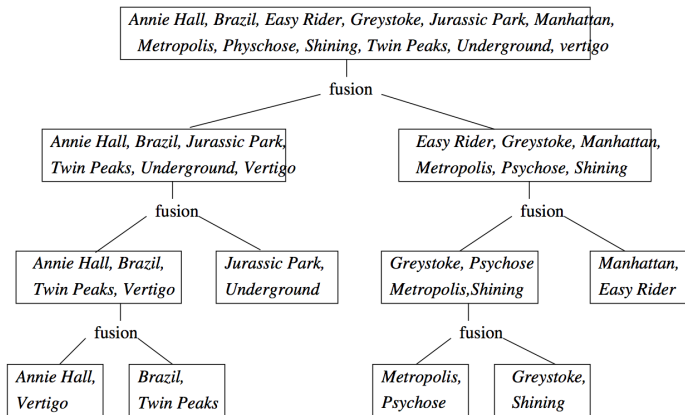
Deuxième phase : la fusion

On groupe les fragments par M (taille de la zone de tri), et on fusionne.



Coût : autant de lectures/écritures du fichier que de niveaux de fusion.

Illustration avec $M = 3$



Essentiel : la taille de la zone de tri

Un fichier de 75 000 pages de 4 Ko, soit 307 Mo.

- $M > 307Mo$: une lecture, soit 307
- $M = 2Mo$, soit 500 pages.

le tri donne $\lceil \frac{307}{2} \rceil = 154$ fragments.

On fait la fusion avec 154 pages

Coût total de $614 + 307 = 921$ Mo.

NB : il faut allouer beaucoup de mémoire pour passer de 1 à 0 niveau de tri.

Avec très peu de mémoire

$M = 1\text{Mo}$, soit 250 pages.

- on obtient 307 fragments.
- On fusionne les 249 premiers fragments, puis les 58 restant. On obtient F_1 et F_2 .
- On fusionne F_1 et F_2 .

Coût total : $1\,228 + 307 = 1\,535$ Mo.

Résultat : grosse dégradation entre 2 Mo et 1 Mo (calcul approximatif).

L'opérateur de tri

L'opérateur de tri est **bloquant**

- Les deux phases du tri-fusion sont effectuées pendant le *open()*
- Le *next()* ne fait que lire, un à un, les tuples dans le résultat du tri.

Conséquence : **latence importante des requêtes impliquant un tri**

Grosse différence entre :

```
| select * from Film
```

et

```
| select * from Film order by titre
```