



Algorithmique

Briques de base - Tableaux

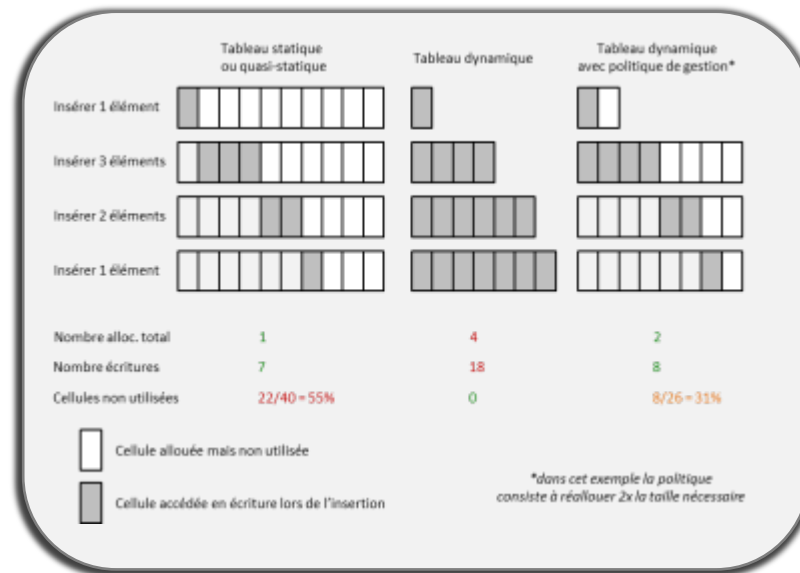


Tableau statique

Statique

Dynamique

Synthèse

- Nombre d'éléments fixe
- Déterminé à la compilation
- Coût d'accès en lecture/écriture constant
- Insertion
 - En fin : coût constant
 - En milieu : linéaire (en fonction du nombre d'éléments à déplacer)
- Limitations évidentes
- Lorsque ces limitations ne sont pas un problème, il faut l'utiliser

Tableau dynamique

Statique
Dynamique
Synthèse

- L'emplacement mémoire est alloué pendant l'exécution de l'application
 - Mécanisme d'allocation dynamique
- Plusieurs méthodes de gestion
 - Quasi-statique : allocation en début d'exécution, une seule fois, en fonction des paramètres d'exécution
 - Gestion dynamique de base : allocation initiale avec une taille faible, réallocation à chaque fois que l'on a besoin
 - Politique d'allocation : on décide d'allouer plus que nécessaire pour réallouer moins souvent













Tableau dynamique



Statique
Dynamique
Synthèse

- Politique d'allocation : impact fort sur
 - Mémoire perdue
 - Coût CPU
- Nécessité de faire un compromis
- Allocation trop importante : gaspillage de mémoire
- Allocation trop faible : effectuée souvent

Synthèse

Statique
Dynamique
Synthèse

| | Tableau statique ou quasi-statique | Tableau dynamique | Tableau dynamique avec politique de gestion* |
|------------------------|--|---|---|
| Insérer 1 élément |  |  |  |
| Insérer 3 éléments |  |  |  |
| Insérer 2 éléments |  |  |  |
| Insérer 1 élément |  |  |  |
| Nombre alloc. total | 1 | 4 | 2 |
| Nombre écritures | 7 | 18 | 8 |
| Cellules non utilisées | 22/40 = 55% | 0 | 8/26 = 31% |

 Cellule allouée mais non utilisée
 Cellule accédée en écriture lors de l'insertion

**dans cet exemple la politique consiste à réallouer 2x la taille nécessaire*