

Analyse du tableau dynamique (vector): fonctionnalités, points forts /faibles

```
#include <vector>

vector<double> tab(5);

tab.size()

tab[i]
```

Un vector est particulièrement efficace pour ajouter/enlever le **dernier élément** (coté **back**) avec **push_back** et **pop_back**.

Cette fonctionnalité permet facilement de mettre en œuvre le concept de *pile* (LIFO = Last In, First Out)

```
tab.push_back(x); // ajoute la valeur x ici
tab.pop_back();   // enlève cette dernière valeur
```

Point faible1: une méthode **insert** permet d'insérer *un ou plusieurs* élément à l'intérieur du vecteur MAIS cela implique un coût de décalage de tous les éléments qui suivent. Même problème avec la méthode **erase** pour enlever *un ou plusieurs* éléments.

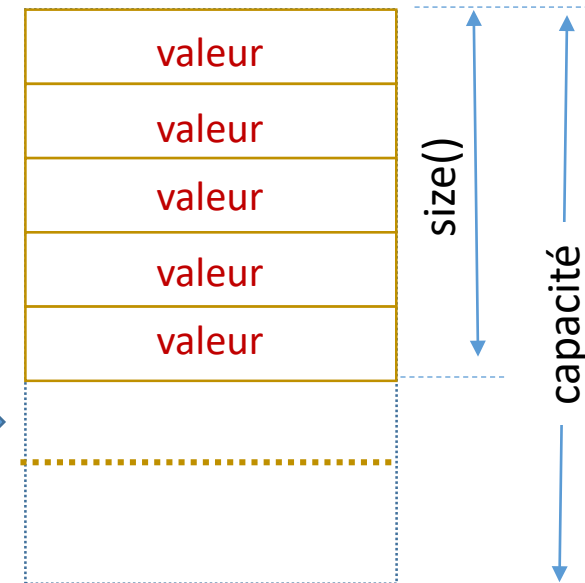
Fait partie des outils de type **container**

Initialisation avec le motif binaire nul

Un vector connaît sa taille

Accès en **O(1)** à tout élément du vector

Extrait du cours Topic12 sem1



Point faible2: il n'existe PAS de méthode pour ajouter/enlever le premier élément (coté **front**). => Utiliser le **container deque** (double-ended queue). Cet outil permet de mettre en œuvre le concept de *file d'attente* (**queue**).

Manipulation sur un tableau dynamique **non trié** (vector)

Complément à la fiche points forts/faibles

Observation1: les méthodes **insert** et **erase** ne doivent pas modifier l'ordre des éléments déjà présent dans un vector. C'est pourquoi elles induisent un coût de décalage des éléments qui suivent dans le vector.

Conséquence négative: coût linéaire => **$O(\text{vector_size})$**

Observation2: si le tableau dynamique n'a ***pas besoin d'être trié***, alors on peut réduire drastiquement l'ordre de complexité des opérations *d'ajout* et de *suppression* d'un élément.

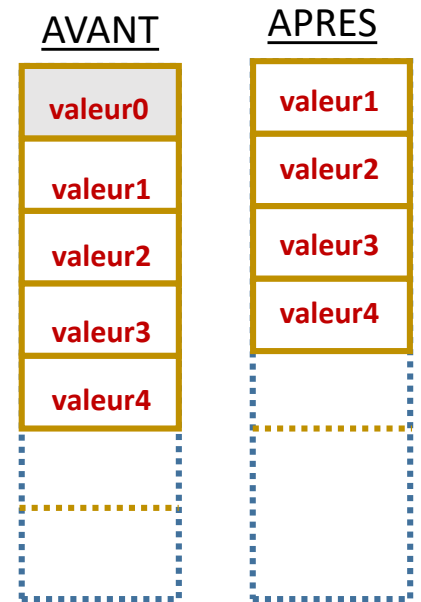
Pour la **suppression**, il suffit d'utiliser la méthode **swap** pour échanger la valeur de l'élément à supprimer avec celle du dernier élément, puis d'appeler **pop_back()**.

Conséquence positive: coût constant => **$O(1)$**

Pour **l'ajout**, il suffit d'appeler **push_back(valeur)**.

Conséquence positive: *généralement* à coût constant => **$O(1)$**

Ex: utilisation de **erase** pour supprimer le premier élément de **valeur0**



Ex: suppression du premier élément du vector **tab** de valeur **valeur0**

