

Héritage et classe abstraite, application aux matrices - 2

Exercice 1 Nous allons modifier la structure interne des matrices creuses : Nous allons utiliser 3 tableaux `val_`, `idx_` et `start_` comme nouveaux membres de donnée. Pour chaque coefficient non nul, la valeur sera conservée dans le tableau `val_`, ordonnée par indice de ligne croissant, puis par indice de colonnes croissant. En même temps, on remplit le tableau `idx_` avec l'indice de la colonnes du coefficient. Les deux tableaux ont donc la même taille. Enfin, le tableau `start_` sera tel que `start_[i]` sera l'indice de début de la i-ème ligne dans les tableaux précédents. Ainsi, le nombre de coefficients non nuls de la ligne i est `start_[i+1]-start_[i]`.

1. Implémenter dans la classe `Matrix_Sparse` la méthode `Matrix full_get() const` qui renvoie la matrice pleine associée à une matrice creuse.
2. Implémenter dans la classe `Matrix` la méthode `Matrix_Sparse sparse_get() const` qui renvoie la matrice creuse associée à une matrice pleine.

Exercice 2 Reprendre le précédent TD en re-écrivant les classes dans le namespace `va` qui implémente les matrices pleines avec un `valarray`, et mesurer la différence de vitesse des multiplications matricielles entre les deux implémentations (utiliser la fonction `gettimeofday()` pour ceci).