

Travaux Pratiques 4

Vous pouvez dans un premier temps sauter les questions précédées de ★.

1. Écrivez une classe `TestIntMatrixMultiplication` permettant de tester et de comparer le temps d'exécution de différents algorithmes pour la multiplication de matrices. Écrivez les méthodes suivantes:

```
# Mesure le temps de algorithm pour la multiplication de matrices
#algorithm est de type IntMatrixMultiplication, a, b et c sont deux matrices
def timeMultiplies(algorithm, a, b, c)
# Affichage d'une_matrice
def_dumbMatrix(m)
```

2. Écrivez une classe `ClassicMatrixMultiplication` implémentant `IntMatrixMultiplication` et reposant sur la formule classique du produit de deux matrices.
3. Implémentez l'algorithme de Strassen (rappel ci-après) pour la multiplication de matrices.

Devez-vous faire une hypothèse sur la taille des matrices que vous allez passer à cet algorithme? ★ Comment la vérifier? ★ Comment traiter le cas des matrices qui ne la vérifient pas?

De quelles opérations élémentaires sur des matrices avez-vous besoin? Écrivez une méthode pour chacune de ces opérations.

Rappel: l'algorithme de Strassen recherche le produit $C = AB$ de deux matrices carrées A et B de taille 2^k , en divisant les trois matrices A, B et C en matrices par blocs de taille égale :

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

où X_{ij} est la sous-matrice carrée de X formée des 2^{k-1} premières (resp. dernières) lignes si $i = 1$ (resp. si $i = 2$) et des 2^{k-1} premières (resp. dernières) colonnes si $j = 1$ (resp. si $j = 2$).

La matrice C est alors déterminée en calculant :

$$P_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$P_2 = (A_{21} + A_{22})(B_{11})$$

$$P_3 = (A_{11})(B_{12} - B_{22})$$

$$P_4 = (A_{22})(B_{21} - B_{11})$$

$$P_5 = (A_{11} + A_{12})(B_{22})$$

$$P_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$

$$P_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

et en remarquant que :

$$C_{11} = P_1 + P_4 - P_5 + P_7$$

$$C_{12} = P_3 + P_5$$

$$C_{21} = P_2 + P_4$$

$$C_{22} = P_1 + P_3 - P_2 + P_6$$

4. Complétez votre classe de test. Réalisez quelques tests comparatifs des deux algorithmes implémentés.

Rappelez la complexité au pire des cas de ces deux algorithmes.

Comment expliquez-vous ce que vous observez?

5. Comment pouvez-vous rendre plus efficace votre implémentation de l'algorithme de Strassen?