

# INF4705 — Analyse et conception d’algorithmes

## TP 1

Ce travail pratique se répartit sur deux séances de laboratoire et porte sur l’analyse empirique et hybride des algorithmes. À la section 3.2 des notes de cours, trois approches d’analyse de l’implantation d’un algorithme sont décrites. Vous les mettrez en pratique pour des algorithmes de multiplication de matrices.

## 1 Implantation

Vous implanterez les algorithmes de multiplication de matrices *conventionnel* et *diviser-pour-régner* (Strassen, section 4.4 des notes de cours). Vous ferez deux versions de ce dernier, avec et sans un seuil de récursivité déterminé expérimentalement par essai-erreur. Les (sous-)exemplaires dont la taille est en deça de ce seuil ne sont plus résolus récursivement mais plutôt directement avec l’algorithme conventionnel. (Nous en parlerons bientôt à la capsule vidéo DpR-capsuleA.)

Vous avez le choix du langage de programmation utilisé mais vous devrez utiliser les mêmes langage, compilateur et ordinateur pour toutes vos implantations. N’oubliez pas de spécifier le cadre expérimental (vitesse du CPU, mémoire vive, ...). Assurez-vous que vos implantations sont correctes en comparant les résultats des trois algorithmes. Le code et les exécutables soumis devront être compatibles avec les ordinateurs de la salle L-4714.

## 2 Jeu de données

Vous trouverez sur le site Moodle du cours un générateur de matrices. Ce générateur prend comme paramètres sur la ligne de commande  $N$  (la taille de la matrice sera  $2^N \times 2^N$ ), le nom du fichier de matrice (par exemple *ex\_8.1* pour une première matrice avec  $N = 8$ ) et le germe du générateur de nombres aléatoires. Le fichier généré débute avec la valeur de  $N$  sur la première ligne et les lignes suivantes correspondent aux lignes de la matrice où chaque nombre est séparé par une tabulation. Voici un exemple pour  $N = 2$  :

```
2
1 3 2 1
0 1 2 2
3 3 3 1
3 0 1 1
```

Pour chaque valeur de  $N$ , générez cinq matrices que vous pourrez multiplier deux à deux, ce qui vous donnera dix exemplaires. Utilisez au moins cinq valeurs consécutives de  $N$  pour votre analyse ; ce choix pourra varier d’une équipe à l’autre selon la qualité de vos implémentations.

### 3 Résultats

Pour chacun des trois algorithmes, mesurez le temps d'exécution pour chaque exemplaire et rapportez dans un tableau le temps moyen par taille d'exemplaire. Vous vous servirez de ces résultats pour l'analyse qui suit. Lorsque vous calculez les temps d'exécution, vous devez séparer le temps de chargement du jeu de test du temps d'exécution de votre algorithme. Vous devrez donc insérer les sondes temporelles à l'intérieur de votre code.

### 4 Analyse

- i) Pour chacun des algorithmes, appliquez le test de puissance. Discutez des résultats obtenus.
- ii) Citez la consommation théorique du temps de calcul pour les algorithmes, en notation asymptotique.
- iii) Pour chacun des algorithmes, appliquez le test du rapport. Expliquez les résultats obtenus.
- iv) Pour chacun des algorithmes, précisez l'analyse asymptotique théorique en calculant les constantes en jeu (troisième approche). Discutez des résultats obtenus.
- v) Discutez de l'impact du choix d'un seuil de récursivité.
- vi) Suite à cette analyse, indiquez sous quelles conditions (taille d'exemplaire ou autre) vous utiliseriez chacun de ces algorithmes. Justifiez.

Note : Pour effectuer vos régressions linéaires, vous pouvez utiliser l'outil de votre choix : Excel, OpenOffice, gnuplot, R (<http://www.r-project.org>, déjà installé au laboratoire), etc. Si vous utilisez des extraits de codes (programmes) trouvés sur Internet, vous devez en mentionner la source, sinon vous serez sanctionnés pour plagiat. (Votre chargé de laboratoire vous encourage à utiliser R ou gnuplot afin d'automatiser la génération de vos graphiques, qui seront nombreux.)

### 5 Remise

*Avant* votre troisième séance de laboratoire, vous devez faire une *remise électronique* en suivant les instructions suivantes :

1. Le dossier remis doit se nommer `matricule1_matricule2_tp1` et doit être compressé sous format zip.
2. À la racine de ce dernier, on doit retrouver un rapport sous format PDF comprenant :
  - une brève description du sujet et des objectifs de ce travail (svp pas de redite de l'énoncé),
  - la description des jeux de données,

- les résultats expérimentaux,
  - l’analyse et discussion
3. Finalement, un script nommé *tp.sh* doit aussi être présent à la racine. Ce dernier sert à exécuter les différents algorithmes du TP. Le fonctionnement du script est décrit à la prochaine section.

## 6 tp.sh

Utilisation

```
tp.sh -a [conv | strassen | strassenSeuil] -e [path_vers_exemplaire]
```

Arguments optionnels :

-p affiche la matrice résultat dans le format des exemplaires, sans texte superflu.

-t affiche le temps d’exécution en ms, sans unité ni texte superflu.

Des exemples de scripts sont disponibles sur Moodle.

## 7 Barème de correction

**1 pt** : exposé du travail pratique

**3 pts** : présentation des résultats

**5 pts** : analyse et discussion

**2 pts** : les programmes (corrects, structurés, commentés, . . .)

**2 pts** : présentation générale : concision, qualité du français