

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 tri.

1 Implantation

Vous planterez les algorithmes de tri par paquets (“bucket sort”) et de tri par fusion (“merge sort”). Vous ferez deux versions de chacun : la première avec un seuil de récursivité de 1 et la seconde avec un seuil déterminé expérimentalement. Les exemplaires dont la taille est en deça de ce seuil ne sont plus résolus récursivement mais plutôt directement avec un tri par insertion.

Vous avez le choix du langage de programmation utilisé mais vous devrez utiliser le même 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 quatre algorithmes. Le code et les exécutables soumis devront être compatibles avec les ordinateurs de la salle L-4714.

2 Jeux de données

Vous trouverez sur le site Moodle du cours les exemplaires à traiter. Les fichiers portent le nom "testset_*n*_*ne*.txt" où *n* représente le nombre d’entiers à trier et *ne* le numéro de l’exemplaire. Par exemple, le fichier *testset_1000_0.txt* possède 1000 nombres et est le premier exemplaire (0) du jeu de tests à 1000 nombres. Pour chaque taille d’exemplaire, trois séries de dix exemplaires (0 à 9, 10 à 19, 20 à 29) sont données. **Vous devez analyser chaque série séparément** : par exemple, les exemplaires 0 à 9 pour $n = 1000, 5000, 10000, 50000, 100000$.

3 Résultats

Pour chacun des quatre algorithmes, mesurez le temps d’exécution pour chaque exemplaire et rapportez le temps moyen pour chaque groupe de dix exemplaires (donc séparément pour chaque série et chaque 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 et chacune des séries d'exemplaires, appliquez le test de puissance. Discutez des résultats obtenus.
- ii) Citez la consommation théorique en meilleur cas, en pire cas et en moyenne du temps de calcul pour les algorithmes, en notation asymptotique.
- iii) Pour chacun des algorithmes et chacune des séries d'exemplaires, appliquez le test du rapport en utilisant chacune des analyses en meilleur cas, en pire cas et en moyenne. Expliquez les résultats obtenus.
- iv) Pour chacun des algorithmes et chacune des séries d'exemplaires, précisez l'analyse asymptotique théorique en calculant les constantes en jeu (troisième approche). Dans chaque cas, choisissez l'analyse (meilleur cas, pire cas, cas moyen) la plus appropriée selon les résultats en (iii). 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 [bucket|bucketSeuil|merge|mergeSeuil] -e [path_vers_exemplaire]
```

Arguments optionnels :

```
-p    Imprime les nombres triés  
-t    Imprime le temps d'exécution
```

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 (incluant la concision) et qualité du français