

Développement en C+

Étude expérimentale de la complexité de quelques algorithmes de tri

Ces TP seront réalisés à l'IUT sous le système d'exploitation Linux.

L'objectif de ce TP (2 séances de 1h30) est d'aborder de façon expérimentale la notion de complexité sur des algorithmes de tris classiques programmés en C+. Pour cela, il faudra tracer des courbes avec l'outil **gnuplot**, chacune d'elles représentant les nombres de comparaisons et d'affectations d'éléments de tableau en fonction de la taille du tableau pour quelques algorithmes de tri.

Question 0 – Découverte de gnuplot

Dans votre répertoire de connexion, créer un répertoire **Dev2** et dans celui-ci, créer un répertoire **Tris** dans lequel vous mettrez tous les fichiers de ce TP. Se placer dans ce répertoire et exécuter la commande `gnuplot` dans un shell.

a) Pour l'utiliser taper dans le terminal de `gnuplot`

```
plot x**2 title "X2", x*sqrt(x) title "x*Racine(x) ", x*log(x) title "x*log(x)"
```

Cette commande permet de tracer les fonctions classiques x^2 , $x*\sqrt{x}$, $x*\log(x)$. Le paramètre `title` permet de définir le nom de la fonction dans la légende

b) Télécharger le fichier `testPlot.txt` depuis le cours Moodle et taper la commande suivante dans le terminal :

```
plot 'testPlot.txt' using 1:2 title "Fonction1", 'testPlot.txt' using 1:3  
title "Fonction2"
```

Cette commande permet de visualiser 2 courbes situées dans le même fichier.

`'testPlot.txt'` indique le fichier contenant les valeurs de la courbe
`using 1:2` indique que la courbe aura comme abscisse la première colonne du fichier et comme ordonnée la seconde colonne.

c) Il est possible de combiner des fonctions provenant de différents fichiers ainsi que des fonctions classiques.

Exemple :

```
plot x*log(x), 'testPlot.txt' using 1:2 title "Fonction1", x**2
```

Question 1 – Récupération des programmes de tri

Télécharger et désarchiver le fichier `tris.tar.gz`. Il contient 3 programmes de tri utilisant l'algorithme du tri à bulles, celui du tri « shell » et celui du tri rapide (tri pivot). Chaque programme remplit un tableau avec des entiers tirés au hasard, l'affiche puis le trie et l'affiche à nouveau une fois trié. Étudier ces 3 programmes pour bien comprendre les algorithmes de tri et regarder également comment on génère les nombres aléatoires. Pour les compiler, utiliser la commande `make`.

Question 2 – Le tri à bulles

Afin d'évaluer la complexité du tri à bulles on va devoir décorer (compléter) le programme qui est dans le fichier `triBulles.C`.

Modifier le code de ce programme de façon à ce que le sous-programme de tri retourne le nombre de comparaisons (`nbC`) ainsi que le nombre d'affectations (`nbA`) nécessaires au tri du tableau.

Pour obtenir des courbes significatives :

- faites varier le nombre d'éléments à trier dans le tableau entre 1 et 300 par pas de 1.
- pour chaque nombre d'éléments, calculer une moyenne de `nbA` et `nbC` sur 100 tests (en changeant le contenu du tableau à chaque fois) et afficher sur une même ligne et séparés par des espaces, la taille du tableau, la moyenne de `nbA` et la moyenne de `nbC`.

Pour sauvegarder ces valeurs dans un fichier et l'utiliser ensuite avec `gnuplot`, pensez à rediriger la sortie de votre programme dans un fichier (opérateur `>` du shell).

Note : utiliser des noms de fichiers différents pour chaque tri (exemple `triBulles.txt`, `triShell.txt`, ...).

Question 3 – Le tri Shell

Même question pour le fichier `triShell.C`.

Dès que les 2 premiers tris seront terminés, visualiser les courbes de ces tris et constater déjà la différence. Exemple de commande `gnuplot` pour comparer le nombre de comparaisons entre les 2 tris :

```
plot 'triBulles.txt' using 1:3 title "Tri à Bulles", 'triShell.txt' using 1:3 title "Tri Shell"
```

Question 4 – Le tri Pivot

Même question pour le fichier `triPivot.C`.

Question 5 – La conclusion ?

Comparer toutes les courbes des différents tris ainsi que les fonctions classiques x^2 , $x \cdot \log(x)$, x , $x \cdot \sqrt{x}$ etc... Enfin des fonctions mathématiques classiques.

Pour sauvegarder des graphiques contenant des comparaisons de courbes dans des fichiers, il faut, avant de les tracer, définir un autre type de format de sortie avec la commande `gnuplot` :

```
set terminal png size 800,600
```

Ensuite, avant de tracer chaque graphique, il faut définir le nom du fichier qui la contiendra avec la commande `gnuplot` :

```
set output 'fichier.png'
```

où *fichier* est à remplacer par un nom de fichier pertinent.

Pour revenir au format normal, utiliser la commande gnuplot :

```
set terminal wxt
```

Après avoir sauvegardé tous les graphiques contenant les comparaisons qui vous semblent pertinentes, rédigez un dossier, créé avec OpenOffice, dans lequel vous inclurez ces graphiques avec vos commentaires et conclusions. Ce rapport devra contenir les noms des deux membres du binôme. Exporter ce rapport au format PDF et le poster, par un seul membre du binôme, dans le devoir Moodle associé à ce TP.

Question 6 – Pour aller plus loin – Le tri par sélection

Ce tri consiste à chercher la position du plus petit élément du tableau et à l'échanger avec le premier élément, puis à recommencer dans le tableau amputé du premier élément, et ainsi de suite jusqu'à avoir trié tout le tableau.

Reprendre le programme de la question 2 en remplaçant le tri à bulle par le tri par sélection.

Pour cela, écrire un sous-programme qui détermine la position du plus petit élément d'une tranche de tableau.

Écrire ensuite un sous-programme de tri par sélection qui utilise le sous-programme précédent.

Comparer ensuite la complexité de cet algorithme avec celle des autres en l'intégrant dans votre rapport.