

TP1 : décompte et représentation graphique du nombre d'opérations

Ce premier TP est l'occasion de faire connaissance avec votre nouvel environnement de travail si vous effectuez vos TP dans l'aile A du bâtiment M5. Nous allons aussi discuter du moyen de compter des opérations dans des programmes et d'afficher le résultat de ces décomptes grâce au logiciel de tracé de courbes Gnuplot.

1 Compiler en Pascal au bâtiment M5

Vous étiez habitué auparavant à utiliser la commande `fpcompile`. Cette commande n'existe plus dans votre nouvel environnement, vous devrez maintenant utiliser la commande `fpc`. Cependant, `fpc` permet de compiler des programmes ... Pour obtenir un comportement similaire à celui que vous aviez avant vous devez ajouter l'option `-Sd`. Ainsi sur la ligne de commande :

```
lxt11 > fpc -Sd monProgramme.pas
```

Pour éviter d'avoir à taper cela à chaque fois, vous pouvez créer un alias, c'est-à-dire associer une commande complexe à un nom.

```
lxt11 > alias fpcompile='fpc -Sd'
```

Maintenant, lorsque vous tapez `fpcompile` c'est la commande `'fpc -Sd'` qui est lancée. Enfin, pour ne pas avoir à créer l'alias à chaque fois, il suffit d'ajouter la ligne ci-dessus au fichier d'initialisation `.bashrc` qui est à la racine de votre répertoire.

Se déconnecter puis se reconnecter afin de vérifier le bon fonctionnement.

2 Compter les opérations dans un algorithme

En Pascal, la façon la plus simple de compter le nombre d'opérations (ou l'espace mémoire occupé) d'un algorithme est d'utiliser des variables globales qu'on n'oubliera pas d'initialiser avant leur utilisation ou entre deux lancements du même algorithme.

Q 1 Programmer une fonction `max` qui calcule la valeur maximale d'un tableau d'éléments passé en paramètre. Vérifier son fonctionnement grâce à la fonction `tableauAleatoire` de l'unité `U_Tableaux`.

Q 2 Déclarer ensuite des variables permettant de compter le nombre de comparaisons d'éléments dans la fonction `max` et modifier la fonction pour réaliser le décompte. Tester.

Q 3 Tester le programme sur des tableaux générés aléatoirement et de différentes tailles en affichant pour chaque tableau le nombre de comparaisons.

Q 4 Modifier le programme pour extraire le nombre de comparaisons minimum, maximum et moyenne pour chaque des tableaux de taille 100. Réalisez des tests sur 1000 tableaux. Pouvait-on s'attendre aux résultats obtenus ?

3 Représentation graphique du nombre d'opérations

On souhaite désormais étudier le comportement asymptotique d'algorithmes.

3.1 Autour de la fonction `max`

Q 5 Modifier le programme de test de la fonction `max` pour qu'il affiche sur deux colonnes la taille du tableau, un espace, le nombre de comparaisons et ce pour des tableaux de taille 1 à 100.

Pour représenter graphiquement le résultat, nous allons utiliser `gnuplot`. Gnuplot permet de tracer des courbes de fonctions mais aussi des suites de points.

La syntaxe pour tracer la courbe d'une fonction est :

```
gnuplot> plot f(x)
```

où `f(x)` est la fonction que vous souhaitez tracer.

Q 6 Essayez de tracer la fonction sinus, la fonction log, le polynôme $x^3 + 4$ (la syntaxe de la puissance est ******).
On peut aussi tracer plusieurs courbes sur le même graphique, en séparant simplement les fonctions par des virgules.

Q 7 Essayez de tracer les courbes des fonction sinus, log, et le polynôme $x + 1$ sur le même graphique.
La syntaxe pour tracer des points est :

```
gnuplot> plot 'fichierDeDonnees' using 1:2
```

Pour tracer les points dont l'abscisse est en colonne 1 et l'ordonnée en colonne 2 dans le fichier `fichierDeDonnees`.
On peut aussi relier les points par une ligne :

```
gnuplot> plot 'fichierDeDonnees' using 1:2 with lines
```

Si vous souhaitez conserver votre courbe dans un PDF, vous devrez spécifier le fichier dans lequel vous allez enregistrer la courbe et le type de fichier.

```
gnuplot> set term pdf
gnuplot> set output 'maCroube.pdf'
gnuplot> plot 'fichierDeDonnees' using 1:2 with lines
```

Pour revenir à un affichage à l'écran :

```
gnuplot> set term x11
```

Vous pourrez trouver une description plus fournie dans l'aide ou sur Internet. Pour obtenir l'aide :

```
gnuplot> help plot
```

Q 8 Lancer le programme de test en redirigeant le résultat vers un fichier. Cela se fait simplement en ajoutant le signe supérieur suivi du nom de fichier à la fin de la ligne de commande :

```
lxt11 $ ./test > nbComparaisons
```

Lancer gnuplot et faire afficher les points représentants le nombre de comparaisons.

Q 9 Ajouter maintenant au graphique le tracé de la courbe d'une fonction représentant la complexité. Quelle fonction est trouvée ? Pouvait-on s'y attendre ?

3.2 Analyse du tri par insertion

Vous avez vu au semestre précédent l'analyse du tri par insertion.

Q 10 Modifier le programme `triInsertion.pas` qui est fourni afin de compter le nombre de comparaisons.

Q 11 Calculer le nombre minimum, maximum et moyen de comparaisons pour des tableaux de taille 1 à 100 (1000 échantillons à chaque fois).

Q 12 Obtenez la représentation graphique du nombre de comparaisons minimum, maximum et en moyenne.