

# L3 Informatique, UE DGINF363 (ALGO6, Algorithmique et modélisation)

## Vendredi 27 janvier : Recherche de motif et algorithme de Karp-Rabin

Ce travail est à faire en binôme. Le compte rendu (un par binôme, indiquant vos **noms et groupes de TD**), est à déposer avant vendredi 27 janvier à 17h [sur la page Moodle du cours](#).

### Objectifs

L'objectif de cette Apnée est d'évaluer les performances de l'[algorithme de Karp-Rabin](#), en comparaison avec un algorithme naïf de recherche de motif.

### Ressources fournies

- [RechercheMotif.java](#), programme à compléter.
- [tests.tar.gz](#), une archive contenant des fichiers de tests

### ► Exercice 1. Compréhension du programme fourni

Le programme Java [RechercheMotif.java](#) contient l'implémentation d'un algorithme naïf de recherche de motif. Ce programme prend en argument une liste de fichiers, chacun contenant deux lignes de texte, une chaîne et un motif.

Lisez (et essayez de comprendre !) le programme RechercheMotif.java.

Vous pouvez compiler ce programme à l'aide de la commande suivante :

```
javac RechercheMotif.java
```

Le programme s'exécute ensuite par la commande :

```
java RechercheMotif
```

Essayez ce programme sur quelques exemples simples (à construire vous-même !).

### ► Exercice 2. Analyse en pire cas

Quel est le coût dans le pire cas de l'algorithme implémenté dans la fonction recherche ? À quel type de données correspond ce pire cas ?

Exécutez le programme sur des fichiers correspondant au pire cas, pour différentes tailles de chaînes. Notez le temps obtenu en fonction de la taille de la chaîne.

### ► Exercice 3. Algorithme de Karp-Rabin

La procédure `rechercheKR` contient une implémentation de l'[algorithme de Karp-Rabin](#).

Convainquez-vous du mieux que vous pouvez que cette implémentation correspond à l'algorithme décrit.

Testez cette implémentation sur plusieurs exemples.

### ► Exercice 4. Évaluation des performances de l'algorithme de Karp-Rabin

Dans certains cas, défavorables pour l'algorithme de recherche naïf, le coût théorique de l'algorithme de Karp-Rabin est en  $O(n+m)$ ,  $n$  étant la longueur du texte et  $m$  la longueur du motif. Il s'agit par exemple des cas où le texte et le motif sont de la forme  $a^*b$  : bien que toutes les sous-chaînes d'indice  $i$  et de longueur  $m-1$  soient un préfixe du motif, la comparaison de caractères ne sera effectuée que pour l'unique sous-chaîne de longueur  $m$  dont le hachage correspond au motif.

Exécutez le programme complété, en utilisant la procédure `rechercheKR`, pour différentes tailles de chaînes. Notez le temps obtenu en fonction de la taille de la chaîne. Vérifiez que vos mesures confirment le coût théorique de l'algorithme de Karp-Rabin.

Sur un même graphique, tracez le temps d'exécution en fonction de la taille de la chaîne pour les deux algorithmes. Que peut-on conclure de ce graphique ?

### Compte-rendu

Le compte-rendu doit être déposé au format PDF. Il devra comporter, **sur 2 pages maximum** :

- une introduction résumant le travail effectué ;
- les réponses à l'exercice 2. ;
- la description des éventuelles modifications apportées au programme ;
- le graphique obtenu à l'exercice 4., ainsi que son interprétation.

Le fichier `RechercheMotif.java` éventuellement complété ou modifié doit être joint à ce compte-rendu.

### Consignes pour le dépôt des fichiers :

- Les fichiers doivent être déposés séparément sur Moodle, et non groupés dans une archive.
- Le nom du fichier du compte-rendu doit contenir le nom des deux étudiants du binôme.
- Les signatures des fonctions fournies ne doivent pas être modifiées.

### Annexe : Gnuplot

Pour tracer les courbes, vous pouvez utiliser le logiciel `gnuplot`, **disponible sur le serveur** `mandelbrot`. On donne ci-dessous quelques indications pour utiliser ce logiciel, sachant que des informations complémentaires sont disponibles [ici](#).

- Format des données : les couples de coordonnées des points à afficher doivent être fournies dans un fichier texte, un couple par

ligne, séparés par un (ou plusieurs) espace(s). Ces coordonnées peuvent être entières ou réelles.

- Le logiciel gnuplot se lance simplement avec la commande `gnuplot` qui a pour effet de démarrer une session interactive. La commande `quit` permettra alors de terminer cette session.
- Une fois une session interactive démarrée (l'invite `gnuplot>` s'affiche) il est possible de tracer la courbe correspondant aux points fournis dans un fichier `data.txt` :

```
gnuplot> plot "data.txt" with linespoints
```

On peut également afficher une seconde courbe dans le même repère, par exemple :

```
gnuplot> replot "data2.txt" with linespoints
```

- Il est possible (et souhaitable...) de nommer les axes et légender les courbes :

```
gnuplot> set xlabel "nom de l'axe x"
gnuplot> set ylabel "nom de l'axe y"
gnuplot> plot "data1.txt" with linespoints title "Légende pour le fichier data1.txt"
          "data2.txt" with linespoints title "Légende pour le fichier data2.txt"
```

- Enfin, il est possible de diriger la sortie de gnuplot dans un fichier image PNG (ou PostScript, ou PDF, ...) :

```
gnuplot> set term png
gnuplot> set out "resultat.png"
gnuplot> plot "data1.txt" with linespoints title "Légende pour le fichier data1.txt"
          "data2.txt" with linespoints title "Légende pour le fichier data2.txt"
```