École Normale Supérieure **2007-2008**
Département d'Informatique

Algorithmique et Programmation
TD n° 3 : Recherche de motifs

Algorithme naïf.

Exercice 1. Supposons que le motif P et le texte T sont des chaînes de caractères de longueurs respectives m et n dont les caractères sont tirés uniformément aléatoirement et indépendamment dans un alphabet à k éléments ($k \geq 2$). Montrer que le nombre espéré de comparaisons de caractères effectuées par l'algorithme naïf de recherche de motif est :

$$(n-m+1)\frac{1-k^{-m}}{1-k^{-1}} \leq 2(n-m+1).$$

Donner également le nombre de comparaisons effectuées par cet algorithme dans le pire des cas et dans le meilleur des cas.

Exercice 2. Supposons que le motif contienne des occurences d'un caractère *joker* pouvant remplacer un mot arbitraire (même le mot vide). Ce caractère joker peut apparaître un nombre de fois quelconque dans le motif, mais n'apparaît pas dans le texte. Donner un algorithme en temps polynomial qui détermine si un tel motif apparaît dans le texte et analyser son temps d'exécution.

Algorithme de Rabin-Karp.

Soit Σ un alphabet. Supposons que chaque caractère $\sigma \in \Sigma$ a un code unique $c(\sigma) \in \mathbb{N}$ (par exemple, son code ASCII). Étant donnés deux entiers p et q fixés, nous définissons la fonction de hachage suivante sur les mots de Σ^* :

$$H(x_1 \dots x_n) = \left(\sum_{i=1}^{n} c(x_i) p^{n-i} \right) \bmod q.$$

Exercice 3. Rappeler l'algorithme de Rabin-Karp qui cherche les occurrences d'un motif P dans un texte T en utilisant la fonction H. Calculez sa complexité dans le le pire cas. Calculer sa complexité en moyenne, en supposant que H répartit équitablement tous les mots sur $\{0, \dots, q-1\}$.

Exercice 4. Soit S un ensemble de motifs. Proposer un algorithme pour chercher toutes les occurrences d'un motif quelconque de S dans un texte T.

Exercice 5. Une *image* est une matrice rectangulaire dont les éléments sont dans Σ. Un *motif 2D* est une famille d'éléments de Σ indexée par un sous-ensemble fini de $\mathbb{Z}^2$ (appelée *forme* du motif). Soient T une image de taille $N \times M$ et soit p un motif 2D de forme $S \subset \mathbb{Z}^2$. Proposer un algorithme qui détermine si P est une sous-motif de T, c'est-à-dire, si il existe $(x, y) \in \mathbb{Z}^2$ tels que $T(x+i, y+j) = P(i, j)$ et $(x+i, y+j) \in [0, N] \times [0, M]$ pour tous les (i, j) dans S.

Automates finis

Exercice 6. Donner l'automate fini pour la recherche du motif suivant : $abcababcac$.

Exercice 7. Considérer l'ensemble des mots $S = \{aba, bab, acb, acbab, cbaba\}$. Donner un automate pour rechercher les occurences d'un mot de S dans un texte. Donner une solution générale à ce problème inspirée de l'algorithme de Knuth-Morris-Pratt et déterminer sa complexité.

Exercice 8. Étant donnés un motif P et un entier $k \geq 0$, décrire comment constituer un automate fini qui détermine les occurences de motifs qui diffèrent de P en au plus k lettres.

Distance de Levenshtein

Exercice 9. La distance de Levenshtein entre deux mots est le nombre minimum d'opérations nécessaires pour transformer un mot en l'autre, où une opération est une insertion, une suppression ou une substitution d'une lettre. Proposer un algorithme qui calcule la distance de Levenshtein entre deux mots.

« Compression d'images »

Exercice 10. Considérons une image en m niveaux de gris (chaque pixel de l'image a un niveau de gris compris entre 0 et $m - 1$). Nous voulons « compresser »cette image en restreignant à n le nombre de niveaux de gris de l'image. Ces n niveaux sont à choisir parmi les m valeurs de l'image d'origine. Dans cette « compression », le niveau de gris de chaque pixel est remplacé par le niveau de gris le plus proche au sein des n niveaux choisis. Décrire un algorithme La problème est le choix des n niveaux de gris qui permettront cette compression avec l'erreur de compression minimale où l'erreur de compression est définie ainsi :

- pour chaque pixel l'erreur est la distance entre sa valeur de niveau de gris et le niveau de gris le plus proche parmi les n niveaux choisis.
- l'erreur de compression est la somme des erreurs des pixels de l'image.

Proposer un algorithme calculant en temps polynomial l'erreur de compression minimale et un choix de n niveaux de gris permettant d'atteindre cette erreur.