Leçon 907 – Algorithmique du texte. Exemples et applications.

2 décembre 2019

1 Extraits du Rapport

Rapport de jury 2017

Cette leçon devrait permettre au candidat de présenter une grande variété d'algorithmes et de paradigmes de programmation, et ne devrait pas se limiter au seul problème de la recherche d'un motif dans un texte, surtout si le candidat ne sait présenter que la méthode naïve. De même, des structures de données plus riches que les tableaux de caractères peuvent montrer leur utilité dans certains algorithmes, qu'il s'agisse d'automates ou d'arbres par exemple. Cependant, cette leçon ne doit pas être confondue avec la 909, ≪Langages rationnels et Automates finis. Exemples et applications.≫. La compression de texte peut faire partie de cette leçon si les algorithmes présentés contiennent effectivement des opérations comme les comparaisons de chaînes : la compression LZW, par exemple, est plus pertinente dans cette leçon que la compression de Huffman.

2 Cœur de la leçon

- Les algorithmes naïfs de recherche de motif et la complexité.
- Les algorithmes non naïfs de KMP et Boyer-Moore.
- Recherche de motif par automate.

3 À savoir

- PLSC, distance d'édition (de Levenshtein).
- Fonctions bordures, notions élémentaires de l'algorithmique du texte (préfixe, suffixe, table des périodes).
- Exemples de mots où les pire cas sont atteints.

4 Ouvertures possibles

- L'algorithme de Karp-Rabin
- Automate de Simon
- Regarder KMP sous l'angle des structures de données ou de l'algorithme (automate de Simon)
- On peut aussi parler de compression de texte et codes correcteurs, mais bien attention à rester dans la leçon.
- Parler des structures (automates, Aho-Corasick, Simon, Suffix Tree, Suffix Trie, Suffix Array, Prefix Trie).
- Recherche approximative de motif.

5 Conseils au candidat

— Bien préciser quels sont les problèmes qu'on cherche à résoudre, les entrées, et les complexités des algorithmes.

— La taille de l'alphabet est un paramètre de la complexité.

— Les dessins, c'est bien.

— Organiser de manière pédagogique les algorithmes. Cela peut vouloir dire par "problème", par "méthode", ou bien par "complexité".

— L'analyse lexicale et syntaxique doit être évitée, bien que techniquement relevant de l'algorithmique du texte.

6 Questions classiques

— Quel algorithme est utilisé par Grep ?

— Faire la table de bordure de tel motif.

— Calculer l'automate des occurrences et le faire tourner sur un exemple.

— Donner un exemple de pire des cas pour les différents algorithmes.

— Pourquoi est-ce que l'automate de Simon possède un nombre linéaire d'arc retours ? Est-il minimal ?

— Quel algorithme vaut-il mieux utiliser pour rechercher plusieurs motifs dans un même texte ?

7 Références

— [*Cro2*] Algorithms on string - Crochemore - à la BU/LSV

— [*Cro*] Jewels of stringology - Crochemore - à la BU/LSV

— [*Han*] Handbook of Theoretical Computer Science - Leeuwen - ?
Contient un bon chapitre par Crochemore *pour l'algorithmique du texte*

— [*Bea*] Éléments d'algorithmique - D. Beauquier, J. Berstel, Ph. Chrétienne - à la BU/LSV
Bonne référence pour l'algo, pleins de dessins et de preuves. Un peu vieillissant et devenu rare.

8 Dev

++ Automate des occurrences - ([Cor], *p.886*) - 907,909,927
Tiens bien en 15 min, mais attention à bien maitriser les petits calculs. Extension possible vers KMP.

++ Plus longue sous séquence commune - ([Cor], *Ch15,4 p341*) - 907,931
Preuve et exemple. Avoir en tête des applications (séquence d'ADN, commande diff)

++ Automate d'Aho-Corasick - ([Cro2], [Bea]) - 907,909,921
Généralise KMP à plusieurs motifs. [Bea] donne bien mieux les intuitions que [Cro2]

+ Calcul de la distance d'édition - ([Cro2]) - 907,931
Bien prendre un coût de 1 *pour chaque opération, quitte à généraliser si le jury pose une question.*