

# Leçon 907 – Algorithmique du texte. Exemples et applications.

9 février 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 Coeur de la leçons

- 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
- 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 structure 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 de motif approximative

## 5 Conseils au candidat

- Bien préciser quels sont les problèmes qu'on cherche à résoudre, les entrées, et les complexités des algorithmes. Par exemple, faire la différence entre la recherche d'un motif, la recherche de  $k$  motifs, la recherche d'une expression régulière
- Pour la complexité, ne pas oublier de considérer la taille de l'alphabet comme un paramètre !
- Il faut faire des dessins en algorithmique du texte, sans quoi on tombe dans des formulations peu claires (et souvent fausses) et des preuves incompréhensibles.
- Faire une simple liste d'algorithmes de texte n'est pas une bonne idée, il faut les organiser de manière pédagogique. Cela peut vouloir dire par "problème", par "méthode", ou bien par "complexité".
- L'analyse lexicale et syntaxique par exemple doit être évitée, bien que techniquement relevant de l'algorithmique du texte.

## 6 Questions classiques

- Quel algorithme est utilisé par Grep ? Libre office ?
- 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 ?

## 7 Références

- **TMP Crochemore TMP**
- **TMP Jewels of Stringology TMP**

## 8 Dev

- **TMP CYK TMP**
- **TMP KMP TMP**
- **TMP Aho-Corasick TMP**
- **TMP Distance d'édition TMP**