(907)

NOM : BALDELLON Prénom : Olivier

N° - Date - Titre de la leçon : 907 : Algo du texte : exemples et applications

Refs : Cormen

Le texte est un format omniprésent, que ce soit pour :

① comme format d'une grande partie de la culture humaine.
② comme interface homme-machine (fichiers sources / UNIX)
③ comme langage de la vie : chaîne d'ADN ou de protéines.

I] Représentation informatique

1) Code Ascii (usage ②)

Sur 7 bits (couramment 8) représente l'alphabet latin, les chiffres, la ponctuation et de symboles divers (retour à la ligne)

Avantage : un texte peut être vu comme un tableau de caractères. Représentation linéaire.

2) UTF-8 (usage ①)

Sur 8, 16 ou 32 bits représente une majorité de glyphes humain.

Avantage : compatible avec l'ASCII

Inconvénient : incompatible avec certains algorithmes

Solution : UTF-32, tableau de caractères.

Remarque : Dorénavant on note Σ l'ensemble des lettres.

3) Compression

Tous les symboles n'ont pas la même fréquence d'apparition.
En français par exemple e ≥ s ≥ a ≥ i ≥ t ≥ n ≥ ...
On peut améliorer le codage en utilisant des tailles variables

Def : Un code binaire complet associe pour chaque noeud soit 0 soit 2 fils.

Un ABC permet de coder des lettres sans ambiguïté : 

Def : Soit un codage donné par un ABC. Soit $\lambda : \Sigma \to \mathbb{N}$ qui à chaque lettre associe la longueur de son codage. Soit φ la fréquence des lettres.
Le coût d'un ABC pour une fréquence φ est donné par

$$\sum_{\sigma \in \Sigma} \varphi(\sigma)\,\lambda(\sigma)$$

Prénom :

NOM :

N° - Date - Titre de la leçon : 904

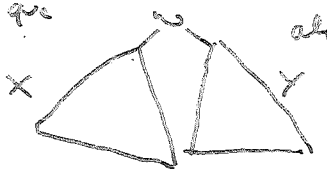
Algo (Huffman) : Trouver l'ABC de coût minimal associé à un texte] Déc 1

II] Dictionnaire

Structure de donnée efficace pour stocker un ensemble de mot (ensemble de taille N)

Représentation naïve : les mots sont les feuilles jusqu'à $|\Sigma|$ fils pour chaque nœud.

Def Un arbre binaire de recherche est un arbre où chaque nœud représente un mot tel que



Inconvénient : Dans le pire cas (peigne) pas mieux que les listes : $\Theta(N)$ avec N nombre de mot.

Def Arbre rouge noir on associe une couleur à chaque nœud : rouge ou noir
- racine est noir
- chaque fils d'un nœud rouge est noir
- chaque branche a le même nombre de nœud noir.

Lemme La longueur des branches d'un ARN est en $\Theta(\log N)$

Algo Insertion, suppression, recherche sont en $\Theta(\log N)$

III] Recherche de Motif

Soit $T[1 \dots n]$ un texte
Soit $M[1 \dots m]$ un mot
Peut on trouver i tel que $\forall k \in [1, \dots, m]$ $M[k] = T[i+k]$

1) Recherche naïve

On fait une boucle sur i et une boucle sur k :
Complexité : $\Theta(m \cdot n)$

2) Rabin - Karp

Principe : * calcul d'un hash de $M[1 \dots m] = h(M[1 \dots m])$
* on pose $h_i = h(T[i+1 \dots i+m])$
* h_{i+1} se calcule en $\Theta(1)$ à partir de h_i.

Algo On compare les h_i avec h et si $h = h_i$ on vérifie que $\forall k \le m$ $M[k] = T[i+k]$

$$h_{i+1} = d\left(h_i - T[i+1]\, d^{(m-1)}\right) + T[i+m+1] \ [q]$$

Prétraitement : $\Theta(m)$
Complexité pire : $\Theta(mn)$
Complexité speré : $\Theta(m+n)$

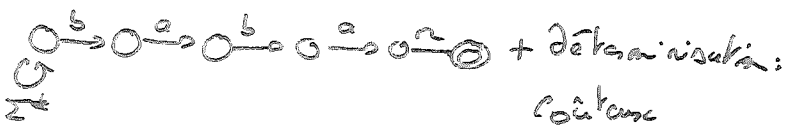
Prénom :

NOM :

N° - Date - Titre de la leçon : 907

3) Automates finis

Version naïve pour chercher "baba"



+ déterminisation : coûteuse

Comment le calculer directement :

Def : un bord de w est un mot à la fois suffixe et préfixe de w

Ex : $\{\varepsilon, a, aba\}$ sont les bords de abadaba

Principe Soit M = abadabac

Si $M[1..7] = T[i+1, \dots, i+7]$

et $M[8] \neq T[i+8]$

alors $M[1..3] = T[i+7-3, i+7]$

car 3 est la taille du plus grand bord de $M[1..7]$

Coût prétraitement $\Theta(m \cdot |\Sigma|)$

Coût exécution $\Theta(n)$

4) Knuth Morris Pratt

On simule l'automate de la partie précédente sans le calculer directement.

Coût prétraitement $\Theta(m)$

Coût exécution $\Theta(n)$

DAC

IV) Compilation et Automate (culture générale)

1) Les langages naturels et autre grammaire peuvent permettre de vérifier la syntaxe d'une entrée.

C = 1+2+3+ ... +9+0 : Date valide → CC/CC/CCCC

2) Les grammaires permettant "d'interpréter" (compiler) interagissent en la transformant en arbre :

"(1+2) X5" →

V) Comparaison de texte

1) Distance d'édition

Algo calcul du nombre d'étape minimale nécessaire pour passer d'un mot w_1 à un mot w_2 en effectuant seulement trois opérations :

- Suppression
- insertion
- substitution

Coût $\Theta(|w_1| |w_2|)$

2) Plus longue sous séquence commune

Algo Une sous séquence d'un mot $a_1 \dots a_n$ est un mot $b_1 \dots b_k$ tel que il existe $\sigma : \mathbb{N} \to \mathbb{N}$ strict croissante vérifiant $\forall i \leq k \quad a_{\sigma(i)} = b_i$

Coût : $\Theta(|w_1| |w_2|)$ pour trouver la PLSSC de w_1 et w_2

Jeux