

Le codage de Huffman

Jake Penney, clarifié et remasterisé par Léon Roux

Septembre 2019

Sommaire

I	Préambule	3
II	Sujet	4
1	Huffman	4
1.1	L'algorithme	4
1.2	L'arbre	5
2	Attentes et limites du Sujet	5

Part I

Préambule

Le javanais ou langue de feu, apparu en France dans la dernière moitié du XIXe siècle, est un procédé de codage argotique utilisant une phonologie parasitaire constituée par l'insertion d'une syllabe supplémentaire entre voyelles et consonnes, dans le but de rendre ce texte moins compréhensible aux non initiés. Cette syllabe comporte un son lié au nom de la variante : " ja " ou " av " dans la variante " javanaise " et une syllabe comportant " f " dans la variante " langue de feu ".

La difficulté réside tout autant dans l'habileté à placer les syllabes supplémentaires de façon naturelle dans la conversation, qu'à la compréhension orale.

Quelques exemples:

- allumettes – > avallavumavettes. Forme complète : avallavumavettaves.
- train – > travailin.
- bonjour – > bavonjavour.
- bicrave – > bavicravave. Forme complète : bavicravavave.
- immeuble – > avimmaveuble. Forme complète : avimmaveublave.
- champion – > chavampavion.
- plans – > plavans.
- supermarché – > savupavermavarchavé.
- poirier – > pavoiravier.
- Peugeot 406 – > Paveugeavot quavatre-cavent-savix.

Vous voila fixés, nous pouvons commencer..

Part II

Sujet

1 Huffman

1.1 L'algorithme

Le codage de Huffman est un algorithme de compression de texte sans perte qui a la particularité d'encoder les caractères en réduisant le nombre de bits utilisés en fonction de leur fréquence, grâce à un arbre appelé *arbre de Huffman*. Le but étant que les caractères les plus fréquents soient codés sur le moins de bits possibles.

L'encodage et le décodage se fait grâce une table de conversion générée à partir de l'arbre binaire de Huffman, qui, lui, tiens compte de la fréquence d'apparition des caractères dans le texte.

Cette façon d'encoder les caractères s'oppose au codage traditionnel ASCII(et son extension, l'Unicode), qui lui, stocke tout les caractères sur 7, 8, 16, 24 ou 32bits sans se soucier de leur fréquence d'apparition.

Exemple: Le mot "*Bonjour*" produit le binaire *100011011100111100* grâce au tableau d'encodage suivant:

Caractère	Binaire
'o'	01
'r'	00
'B'	100
'n'	101
'j'	110
'u'	111

et peut être à son tour décodé comme cela:

100	01	101	110	01	111	00
B	o	n	j	o	u	r

1.2 L'arbre

Mais comment construit-t-on ce tableau d'encodage ? La reponse est dans le titre.

2 Attentes et limites du Sujet

Dans ce sujet, on vas vous demander d'implementer cet algorithme en

- Python ≥ 3.0