2019 2020 année des mathématiques

Mathématiques et numérique
Compression de données

Le codage de **Huffman** est une méthode de compression statistique de données qui permet de réduire la longueur du codage d'un alphabet. Le principe du codage de Huffman repose sur la création d'une structure d'arbre composée de nœuds. On observe ainsi des réductions de taille de l'ordre de 20 à 90% par rapport au codage ASCII.

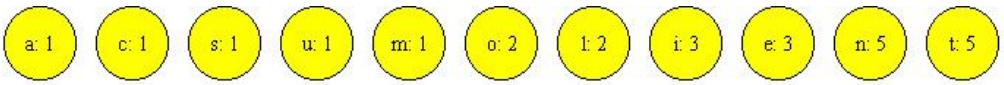
Supposons que notre fichier soit extrêmement simple, et constitué d'un mot unique :

anticonstitutionnellement

Il y a 25 caractères dans ce fichier ; chaque caractère étant codé par un octet de 8 bits (codage ASCII) cela signifie donc 25 octets, ou encore 200 bits.
Voyons ce que nous pouvons faire de cela.

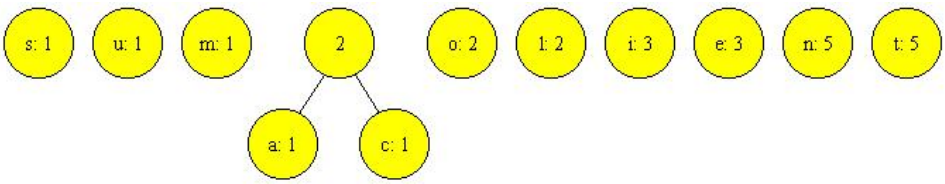
Lettres	a	c	s	u	m	o	l	i	e	n	t
Effectifs	1	1	1	1	1	2	2	3	3	5	5

Étape 0 :
On commence par créer « les nœuds racines » de l'arbre de codage en fonction des entrées du tableau.
À chaque lettre, on associe un poids égal à son nombre d'occurrences.
On classe les nœuds dans l'ordre croissant :

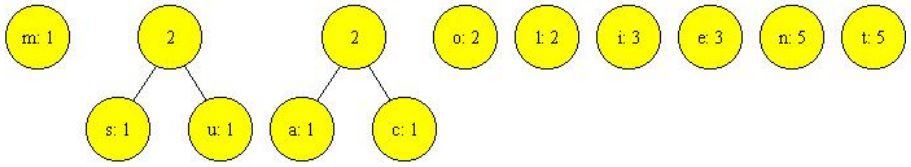


On commence l'itération : à chaque fois on supprime les deux arbres de gauche et on les remplace par un "arbre somme". Le nouvel arbre est inséré dans la liste en respectant l'ordre croissant, et on recommence jusqu'à ce qu'il ne reste plus qu'un seul arbre :

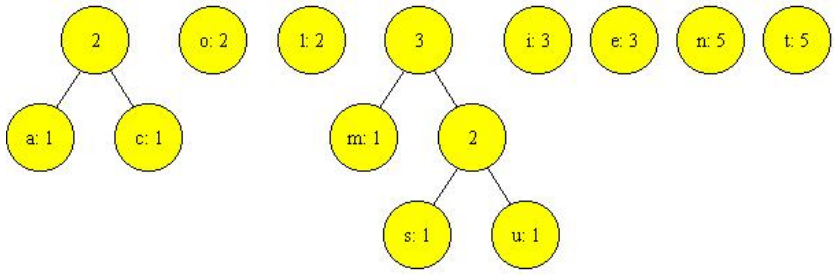
Étape 1 :



Étape 2 :

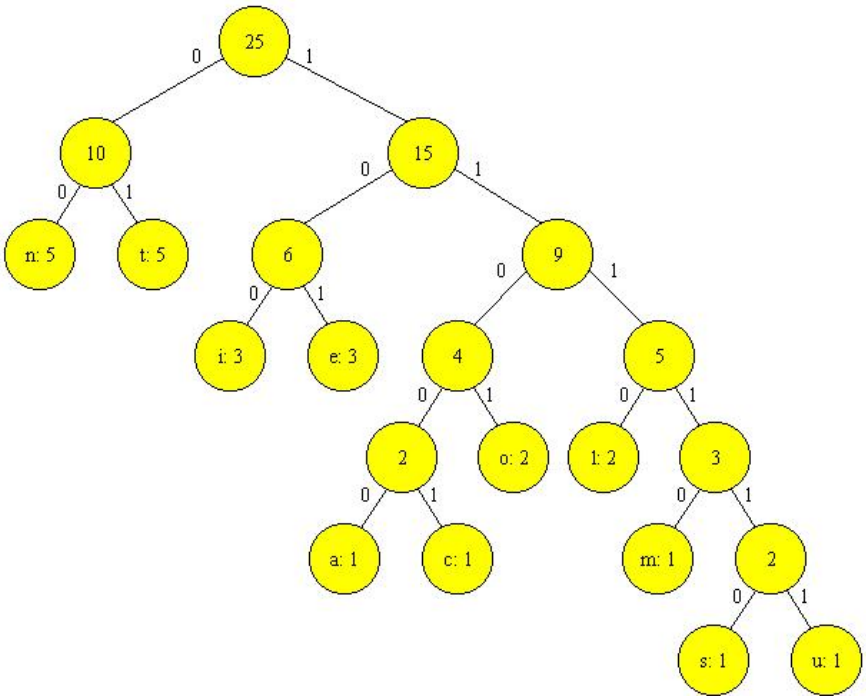


Etape 3 :



etc ...

L'arbre final :



Maintenant, le code associé à chaque Lettre n'est autre que le chemin d'accès au nœud terminal correspondant, depuis la racine, en notant 0 la branche de gauche et 1 la branche de droite.

Finalement :

Lettre	n	t	i	e	a	c	o	l	m	s	u
Code binaire	0	1	100	101	11000	11001	1101	1110	11110	111110	111111

Et voici maintenant, transcrit avec notre nouveau code, le mot de départ :

11000000110011001110100111110011000111111101100110100001011110111010111110101000 1

ce qui nous fait 81 bits, au lieu de 200 au départ! Cela correspond à un taux de compression de 60 %