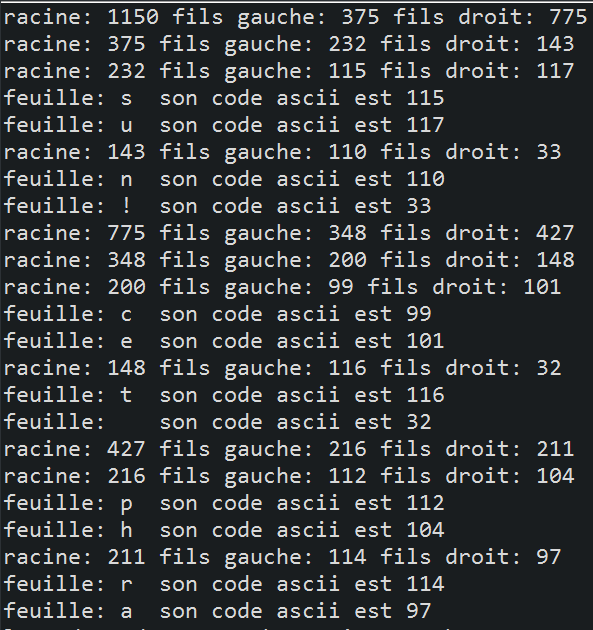
**Solution Naïve :**

Le problème qui se pose est de pouvoir transformer un texte donné en une série de 0 et 1 que l’ordinateur peut comprendre. La première solution qui saute aux yeux est d’associer à chaque caractère un code binaire de 8bits (avec 8bits on peut coder 2^8 = 256 qui est un nombre suffisant pour représenter les caractères spéciaux de plusieurs langues).

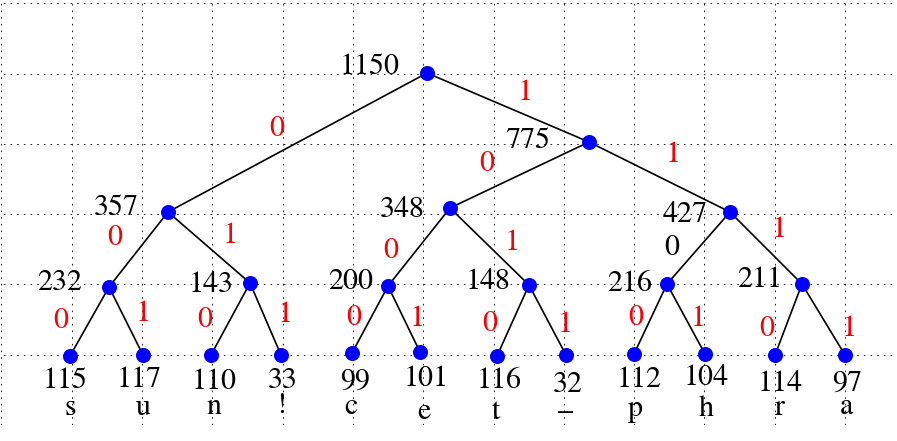
Pour que l’ordinateur nous génère une liste des caractères qui existent dans le texte donné en input et les codes binaires correspondants, on va créer un arbre binaire strict (dont tous les nœuds possèdent zéro ou deux fils).

Il est construit en initialisant tout d’abord toutes les caractères présents dans la phrase qu’on veut coder à des feuilles (racine : le code ASCII du caractère, fils gauche: null, fils droit: null). Ensuite on somme les valeurs des feuilles deux à deux et on les affecte à une racine qui a comme fils gauche la valeur du première feuille et comme fils droit la valeur du deuxième feuille.

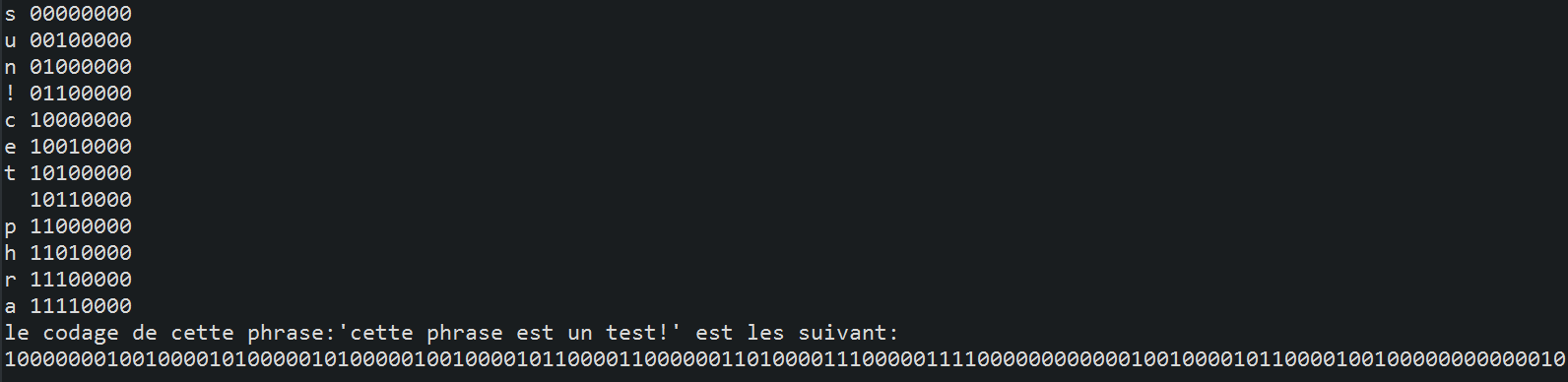
Voici l’arbre que construit notre code qui on lui donne en input la phrase suivante : « cette phrase est un test ! »



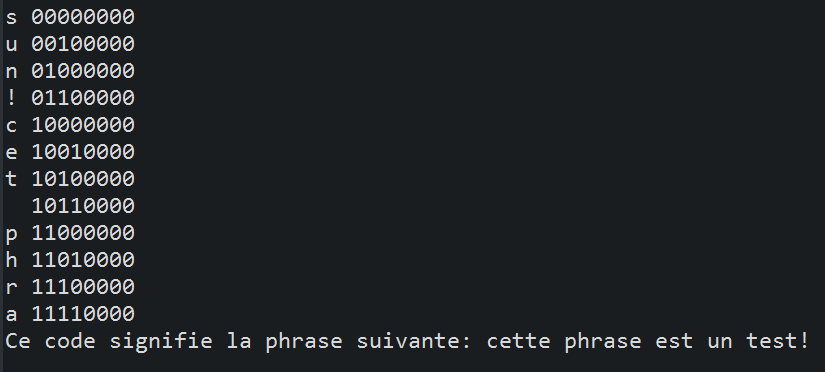
Maintenant, partant de la racine, on associe à chaque fils gauche un « 0 » et à chaque fils droit un « 1 ». Pour mieux comprendre, voici une représentation de l’arbre que j’ai réalisé à l’aide d’un outil appelé géophar :



Pour obtenir le [code binaire](https://fr.wikipedia.org/wiki/Syst%C3%A8me_binaire) de chaque caractère, on remonte l'arbre à partir de la racine jusqu'aux feuilles en rajoutant à chaque fois au code un 0 ou un 1 selon la branche suivie.



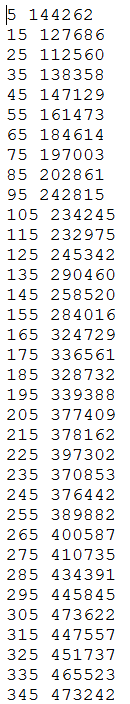
Quant au décodage il faut que la personne qui décode ait la même liste des fréquences et procède au même algorithme pour construire l’arbre. Ensuite il peut générer un tableau qui contient les caractères et leurs codes. C’est ainsi qu’il peut déchiffrer le code reçu.



En ce qui concerne la complexité, on a créé une fonction qui prend en input la longueur du mot souhaité et génère ensuite un mot qui se compose de caractères aléatoires et donne ensuite son code binaire et le temps d’exécution du code. On répète cette opération 50 fois pour la longueur du mot pour ainsi calculer la moyenne des temps d’exécution. Dans un tableau qui doit contenir la longueur du mot et son temps d’exécution, on stocke cette moyenne calculée. Ensuite on augmente le nombre de caractères et on répète 50fois encore, et on procède ainsi 100fois.

Le tableau qu’on obtient à la fin va être utile pour tracer la courbe de complexité.

Extrait du tableau :



La courbe de complexité :